DISSERTATION

Optimal Allocation and Scheduling of Demand in Deregulated Energy Markets

Submitted to the Postgraduate School of the
Faculty of Informatics,
Vienna University of Technology,
in partial fulfillment of the requirements
for the degree of Doctor of Technical Sciences

Under supervision of

Prof. Dr. Dietmar Dietrich
Institute of Computer Technology
Vienna University of Technology

and

Prof. Dr. Nicholas R. Jennings
School of Electronics and Computer Science
University of Southampton

by

Yoseba Koldobika
Peña Landaburu

Student Id: 9900186
Getreidemarkt 13/31
A1060, Vienna, Austria
yoseba.penya@wu-wien.ac.at

Vienna, March 2006 . . . .

Abstract

The deregulation of the electricity industry in many countries has created a number of marketplaces in which producers and consumers can operate in order to more effectively manage and meet their energy needs. To this end, this thesis develops a new model for electricity retail where end-use customers choose their supplier from competing electricity retailers. The model is based on simultaneous reverse combinatorial auctions, designed as a second-price sealed-bid multi-item auction with supply function bidding. This model prevents strategic bidding and allows the auctioneer to maximise its payoff. Furthermore, we develop optimal single-item and multi-item algorithms for winner determination in such auctions that are significantly less complex than those currently available in the literature. Nevertheless, the consumption of the energy of each singular auctioneer has to be regulated and adapted to the submitted bids in order to maximise the payoff. To this extent, this work models the constellation of energy consuming devices as a distributed constraint optimisation problem (dCOP). However, to operate within this problem domain, dCOP algorithms must be complete, work with global cost functions and complete solutions, and, currently, there are no dCOP algorithms that fulfil these conditions. Therefore, this thesis develops a novel optimal dCOP algorithm, called COBB (constraint optimisation by broadcasting), and, additionally, adapt state-of-the-art counterparts to our domain. Empirical comparisons show that COBB clearly outperforms all of them. Finally, this work outlines effective strategies to reduce the network overload caused by broadcasting to maintain it into a range where it is a reasonable trade-off for COBB's efficiency.

Contents

In the field of science, however, the man who makes himself the impresario of the subject to which he should be devoted, and steps upon the stage and seeks to legitimate himself through "experience", asking: How can I prove that I am something other than a mere "specialist" and how can I manage to say something in form or in content that nobody else has ever said ? –such a man is no "personality". Today such conduct is a crowd phenomenon, and it always makes a petty impression and debases the one who is thus concerned. Instead of this, an inner devotion to the task, and that alone, should lift the scientist to the height and dignity of the subject he pretends to serve. And in this it is not different with the artist. In contrast with these preconditions which scientific work shares with art, science has a fate that profoundly distinguishes it from artistic work. Scientific work is chained to the course of progress; whereas in the realm of art there is no progress in the same sense. It is not true that the work of art of a period that has worked out new technical means, or, for instance, the laws of perspective, stands therefore artistically higher than a work of art devoid of all knowledge of those means and laws–if its form does justice to the material, that is, if its object has been chosen and formed so that it could be artistically mastered without applying those conditions and means. A work of art which is genuine "fulfilment" is never surpassed; it will never be antiquated. Individuals may differ in appreciating the personal significance of works of art, but no one will ever be able to say of such a work that it is 'outstripped by another work which is also "fulfilment".

M. Weber, *"Science as Vocation"*

Prologue

The following statement may sound as an Oscar®-winner thanks-giving talk but indeed, I didn't expect this. I mean, writing a PhD. Never, in younger days, did I think of pursuing an academic career, teaching... or doing the kind of stuff I do now.

When I was a child, my "when I grow up" dream was to become a carpenter-astronaut (which was a kind of trade-off between some of us since my best friends' family had a long tradition in carpentry). Many years after that, I chose Computer Science simply not to study Laws or Medicine and, well, in my last term I got the opportunity of studying abroad. Thus, quite unexpectedly, I ended up in Vienna for three months with an Erasmus grant. I liked it so much that I prolonged my stay to six months and finally got a position at the Institute of Computer Technology (ICT), where I was writing my Diploma Thesis (having found the announcement by sheer chance when I already had another topic).

At the beginning of my research there, the idea of a PhD was a kind of nebula: something vague, unprecise to be finished someday. I was too entangled in my own life as to really take care of such a thing in the future. An then, September 2001 came. It was not the 11th, but some weeks after, in the marvelous St. Petersburg. I was working there for a week in order to prepare and hold a meeting with the Russian partners of the project I was leading -remote energy metering and billing.

Peter Palensky, who was a kind of model for me at the ICT, was also in town for some lectures so after a big Friday-night party, we spent the day after strolling in the Champ-Elysées-alike “Petergof” (Petershof). In those talks with Peter, was were I started to believe in the existence of the road that ends today. And, in this process (as in Kavafis’s poem), the important thing is the course itself, not the finish. Thus, hereby I say goodbye not only to the research of four years but also to four very intense years of my life.

Thesis Structure

This dissertation presents the following arrangement.

- **Chapter 2** provides the required theoretical background. We detail aspects concerning constraint problems and specially, distributed constraint satisfaction and optimisation algorithms. Further, we discuss topics from Game Theory, specially those related to Auction Theory. Finally, we explain the motivation of the Deregulation of European Electricity Markets, comment on its current status, and then go on to connect this to DSM. Every section includes an account of respective relevant related work to this Thesis.

- **Chapter 3** describes the overall architecture of the system. We outline the relationships between the two compounding parts of the system, detail the nature of the agents participating in the design and provide an introduction to the two subsequent sections, where we deal with higher detail with each of the two parts.

- **Chapter 4** details the optimal allocation of demand. We list the requirements stemming from our particular problem setting and detail the addressed market design. Further, we present the single-item and the multi-item clearing algorithms for this marketplace and analyse their complexity and optimality. Finally, we consider different ways on improving the efficiency of the algorithms and evaluate all the algorithms against the only existing counterpart.

- **Chapter 5** details the optimal scheduling of demand. We present a taxonomy of consumers according to their suitability to take part into a DSM-system. We delineate the features of this DSM-system modeled as a distributed constraint optimisation problem. Then, we detail a novel algorithm to solve this problem and the tailoring of counterparts to operate within our problem domain. Finally, we evaluate all the algorithms, analyse their behaviour along different dimensions and show that the novel algorithm outperforms all the rest.

- **Chapter 6** summarises and discuses the contributions of this dissertation and outlines the avenues of future work in the confluence of energy management, auction theory and distributed artificial intelligence.

Acknowledgements

There are many people I would like to thank but I will start with the person who had the biggest influence and importance in this dissertation: Prof. Nick Jennings. He really is as you expect him to be before you have met him. Working at his Lab in Southampton was an unforgettable experience, both scientifically and personally. His team received me warmly and I learned a lot: Raj Dash (congrats!), Alex Rogers, Viet Dang, Etty David, Xudong Luo, Minghua He and the always patient Jane Morgan. And, specially, Nadim Haque, who helped me in Compiègne and in England to find a proper solution for the overbooking problem (see Chapter 4). Thank you all!!!

In Vienna, Prof. Dietrich gave me the possibility of writing this PhD and the freedom of choosing a topic that was very interesting for me. It has been a difficult way together and I appreciate all that he has taught me, specially, these last months of hard work together with the manuscript. Moreover, I'm also grateful to Peter Skiczuk, my advisor of the diploma thesis and the person who backed me for the free position at the ICT. Likewise, Peter Palensky and Prof. Thilo Sauter, my project-leaders during the time at ICT, patiently helped and promoted me. And, impossible to forget, my colleagues and mates Gerhard Pratl and the Russian-connection: Alexej Bratoukhine and Maxim Lobashov. It was a fantastic time together.

When I changed to the Vienna University of Economics and Business Administration, Prof. Gustaf Neumann and Jan Mendling provided me with useful comments and advices on the second part of the PhD (Chapter 5) and Uwe Zdun helped me tame LaTeX.

I would also like to thank my students David Spanberger and Hannes Stauss for their work and help implementing qAdopt (see section 5.5) and Natascha Ljubic with whom section 2.3 was partly co-written.

Then, Hanna (*hdl!*) encouraged me to finally start writing and provided me with the perfect environment to succeed in this last challenge of the process. Finally, I cannot forget my mate José Avendaño Córcoles for giving me all his friendship, comprehension and support throughout all these years: *¡Gracias, Pepis!*

And, last but not least, my family. My Grandmother Isabel set the example for me and this PhD completes our agreement: *Tata, he cumplido la promesa*. All my relatives, starting by Aita and Ama, gave me their support, love and strength so I was able to continue the course and finally get to the shore of this first Ithaka of mine. This work is dedicated to all of them.

Eskerrik asko danoi, bihotz-bihotzez!

Yoseba Penya
March 2006

1 The world is all that is the case.
1.1 The world is the totality of facts, not of things.
1.2 The world divides into facts.
2 What is the case–a fact–is the existence of states of affairs.
2.1 We picture facts to ourselves.
2.2 A picture has logicopictorial form in common with what it depicts.
3 A logical picture of facts is a thought.
3.1 In a proposition a thought finds an expression that can be perceived by the senses.

L. Wittgenstein, *"Tractatus Logico-Philosophicus"*

Chapter 1

Introduction

The chapter opens with the motivation of this PhD and describes the current situation in the European electrical market. With this environment in place, it continues to account the achievements and contributions of this work to the state of the art and finalises detailing the structure of the thesis.

1.1 Motivation

The deregulation of electricity markets began in the early nineties when the UK Government privatised the electricity supply industry in England and Wales [PJ05]. This process has been subsequently followed in many other countries. In most cases, the restructuring involves separating the electricity generation and retail from the natural monopoly functions of transmission and distribution. This, in turn, leads to the establishment of a *wholesale electricity market* for electricity generation and a *retail electricity market* for electricity retailing. In the former case, competing generators offer their electricity output to retailers and in the latter case end-use customers choose their supplier from competing electricity retailers.

Here we focus on retail markets, which differ from their more traditional counterparts because energy cannot be stored or held in stock (as tangible goods can).

Consequently, retailers are forced to work with consumption prognoses, which, in turn, creates a number of risks. First, producing more than is consumed is not economical. Moreover, the price of the energy mainly depends on the production cost and this typically rises with the amount of energy produced. Second, if the demand exceeds the prediction, suppliers must find additional energy to avoid a blackout. Finally, there are non-negligible costs stemming from the variation in the electricity production volume that most of the traditional types of energy generators (e.g. hydroelectric, thermoelectric, nuclear) have to face.

In this way, the desideratum is to achieve a market model where retailers have the most accurate possible prognosis and the capability of influencing or guiding customers' consumption. To this end, there have been a number of initiatives, grouped under the general banner of *Demand-Side Management* (DSM), whose main objective is to distribute the demand over time to avoid peak loads.

DSM has been promoted by Utility Companies (UCs) as an alternative to building new power plants. Its (deliberately infeasible) objective consists of smoothing demand so that ideally it is a flat constant energy consumption 24 hours a day, 365 days a year. This profile embodies the ideal circumstances for energy producers and Transmission System Operators (TSOs) since the former may employ cheap and stable production methods, harmless to the environment, and the latter control a transmission grid with constant load [Pal01].

An analysis of the current electricity market yields two main observations. First, it is still far from deregulated, as we will see. And second, the employed DSM-techniques do not take advantage of the new conditions induced by deregulation. Let us go into more detail with these issues. Nowadays, retailers must face the risk of working with demand estimations [Mel01] and most customers only partially enjoy the benefits of the deregulated market. They typically sign mid-term contracts with a single supplier and tariffs do not reflect the pressure of competition. Moreover, whereas classical capitalist pricing policies encourage demand by applying discounts on quantity (the more you buy, the cheaper the unit price becomes), actual electricity contracts often include a threshold above which the consumption becomes more expensive. In this way, we want to move to a more dynamic environment where the benefits of deregulation can be fully achieved.

Now, the easiest way to accomplish this goal is by setting the price of the energy depending on the actual demand load. Thus, the higher the demand, the more expensive the price, and vice versa. Based upon these premises, many utility companies (UCs) already present a basic form of DSM by offering a cheaper night tariff. Our aim in this work is to improve and extend this simple market model to permit UCs to express more complex aims and, thus, increase their influence on customers. For instance, in order to lighten the peak-time load, the supplier can offer a discount for consuming a small amount of energy at 8 am (peak-time) and a larger amount at midnight (off peak). This incentivises the customer to reschedule some tasks to midnight (e.g. the dishwasher or the washing machine). If many clients accept this compromise offer, the UC will have achieved a double goal. It will have a more accurate prognosis for 8 am and midnight and it will also have shifted some of the peak-time consumption to off peak. In e-commerce terms, this process can be seen as a *reverse combinatorial auction*.

It is "reverse" because the customers pick one of the available companies and tariffs to supply their future consumption. And it is "combinatorial" because bidding for a bundle of items is typically valued differently from bidding separately for each of the constituent items (e.g. the *combination* of consuming at 8am and midnight is more appreciated, and thus rewarded, than, for instance, the combination of consuming at 10am and 11am).

While combinatorial auctions provide very efficient allocations that can maximise the revenue for the auctioneer, their main drawback is the complexity of the *clearing* process in which buyers and sellers are matched and the quantities of items traded between them are determined. Specifically, clearing combinatorial auctions is non-deterministic polynomial-time hard (NP-hard) [FLBS99]. Moreover, most work in this area deals with clearing combinatorial auctions with *atomic propositions* [San02]. Thus, bids are either accepted or rejected in their entirety, which may limit the profit for the auctioneer. A more efficient solution is to allow bidding with demand/supply functions [SS01, DJ03], in which bidders submit a function to calculate the cost of the units to be bought or sold. This allows the customer to accept parts of different bids and constitutes a powerful way of expressing complex pricing policies. In our case, production costs can be easily reflected in the supply function and if bids are accepted partially, there may be more than one winner for the same auction and item. This enables customers to accept different parts of bids from different bidders so they can get energy simultaneously from several suppliers. Since the transmission and distribution grids are shared and the path followed by the electricity cannot be tracked down, it is impossible to determine the producer of the energy being consumed. Therefore, the hypothesis of customers being simultaneously supplied by several UCs does not pose any technical problems.

Nevertheless, it does not help if we design a very sophisticated market that optimally allocates demand among suppliers and maximises participant's payoff if customers do not vary their consumption or, more accurately, if the devices of each single customer do not adapt their demand according to market's laws (i.e. implicitly, the bids submitted to the auctions described above). Assuming that devices issue consumption prognoses, any effective DSM method would require that some of these devices are able to modify their consumption ad libitum. This is, they may anticipate or postpone an energy consuming task if this is the best for the overall system and what they want to do.

In this way, if we want to find an optimal solution (meaning the cheapest placement possible for each task), we have to process all possible combinations of each task at a different consumption alternative and then select the best one (say cheapest), according to the submitted bids. If we see these consumption alternatives as possible values of a variable and, additionally add some constraints between variables and use the price of the consumed electricity as result of the cost function, then we have set it out as a constraint problem or, more specifically, a constraint optimisation problem. The possible alternatives devise a solution space of combinatorial dimensions (i.e. depends on the number of devices and the number of alternatives of each appliance) that must be traversed in the quest for a better solution.

Moreover, security and privacy issues (if we address not a single household but a number of them) on the one hand, and compatibility and inter-operability issues in the consumption alternatives elicitation, on the other, indicate we should solve this problem in a distributed manner. So it becomes a *distributed* constraint optimisation problem (dCOP). Hence, the components of the system have to iteratively test all possible combinations in the solution space in order to identify and adopt the most convenient one, according to the bids submitted to the aforementioned auction and to the consumption alternatives of every single device in the system. Unfortunately, existing dCOPs algorithms are not applicable to our problem domain for a number of reasons (as we will detail in Chapter 5).

Thus, motivated by the new opportunities offered by deregulation, we deal in this dissertation with a twofold main goal. On one side, the design of a new marketplace that enables us to optimise the allocation of demand. On the other side, the development of a new method that optimises the scheduling of that demand. Both issues demarcate separated, but interrelated and interconnected domains.

Against this background, we address hereby a combination of Game Theory (specially Auction Theory) on the one hand, and Distributed Artificial Intelligence on the other, to optimise allocation and scheduling of demand in deregulated markets. First, we have designed a system of *simultaneous reverse combinatorial auctions* that assures maximum benefits for the auctioneer (and thus, optimal demand allocation, as we will see), and the corresponding optimal clearing algorithms. Second, we have developed a novel *distributed constraint optimisation problem algorithm* that efficiently finds the best demand schedule according to the supply bids received in the aforementioned market.

Moreover, both the marketplace and the dCOP environment are embodied by two respective multi-agent systems (sharing one common element, as explained in chapter 3, that acts as communication pipe between the separated systems). Classical reasons for choosing MAS technology are pointed out by Palensky [Pal01], quoting previous work of Jennings et al. in [JCL$^+$96] and[CLJ96]:

- *Economy*: The reuse of existing software is cheaper than developing new software.
- *Robustness*: The overlapping expertise of the single agents makes the overall system more fault-tolerant.
- *Reliability*: The same overlapping expertise makes the system more reliable because information can be cross-referenced and compared.
- *Natural representation of the domain*: The system represents the way the control engineers work when a disturbance occurs. There are specialised agents that have to cooperate to get all the information and to save the situation.

More specifically, we would add the following concrete issues, directly related to the challenges that our problem setting poses. First, agents are *autonomous*. This property enables agents to operate within the decentralised control regimes of our problem domain: a distributed marketplace and a distributed constraint optimisation system. Second, agents are *reactive*. This property enables them to rapidly respond to changes, a property required specially in our dCOP system. Finally, they are *flexible*, which enables them to cope with the dynamism and fast-changing conditions of both the market place and the dCOP system[1].

1.2 Research Contributions

The following account constitutes the list of objectives of this work that will allow its contributions to be judged.

[1] The properties of agents accounted here were first detailed by Wooldridge and Jennings in their seminal work [WJ95].

- **DSM Model**: Develop an optimal DSM-model integrating electricity retailers and consumers, adapted to the new conditions posed by deregulated markets.
- **Demand allocation**: Develop an optimal demand allocation model that assures maximum benefit for clients and, simultaneously, provides suppliers with a tool to influence their consuming behaviour.
- **Demand scheduling**: Develop an optimal demand scheduling model that assures finding the best possible consumption alternative for each device, according to the cost of the actual electricity.

Hence, we can describe next the contributions to the state of the art made by this thesis, together with the works in which they were published.

- First, we present, for the first time, a novel **energy retail market** designed as a system of reverse combinatorial auctions with supply function bidding. This novel market, introduced in [PJ05], allows customers to increase their profit and provides UCs with a mechanism to influence customers' behaviour.
- Second, in the same work we develop new optimal **clearing algorithms** tailored to electricity supply functions that perform better than the existing more general clearing algorithms.
- Third, in [PPL03] we provide the first taxonomy of electrical devices according to their DSM-ability (i.e. the degree of participation within a DSM-system).
- Fourth, we introduce the first **DSM-system** of devices modeled as a distributed constraint optimisation problem (dCOP), as presented in [PJN05].
- Fifth, in the same work we develop, for the problem stated above, a new optimal **dCOP algorithm** and tailor existing counterpart algorithms to solve it. We compare these and show how our novel algorithm outperforms the rest.
- Last but not least, we describe the **integration** of both marketplace and consumers' systems into an architecture that exploits the benefits of deregulated electricity markets to optimally allocate and schedule demand.

1.3 Summary

This dissertation addresses a novel market design in order to take advantage of the new situation that the deregulation of European (and other countries') electricity markets pose. Customers auction their future energy demand and utility companies compete to provide this. This market model establishes a trade-off where clients become cheaper energy in exchange for predictable demand. Still, redesigning the market format is not enough if clients lack a method to control their electricity consumption. To this end, this thesis describes a device demand optimisation way, so devices adjust their demand to the needs of the rest customer's devices and achieve the cheapest possible consumption profile according to the offers and discounts submitted by the utility companies.

Before accurately describing the research done in both, the novel market design and the demand optimisation method, let us start with an introduction to the three main topics converging in this dissertation: Constraint Problems, Game and Auction Theory, and Energy Markets. In this way, Chapter 2 will help more easily comprehend the work described afterwards.

All language is a set of symbols whose use among its speakers assumes a shared past. How, then, can I translate into words the limitless Aleph, which my floundering mind can scarcely encompass? Mystics, faced with the same problem, fall back on symbols: to signify the godhead, one Persian speaks of a bird that somehow is all birds; Alanus de Insulis, of a sphere whose center is everywhere and circumference is nowhere; Ezekiel, of a four-faced angel who at one and the same time moves east and west, north and south. (Not in vain do I recall these inconceivable analogies; they bear some relation to the Aleph.) Perhaps the gods might grant me a similar metaphor, but then this account would become contaminated by literature, by fiction. Really, what I want to do is impossible, for any listing of an endless series is doomed to be infinitesimal. In that single gigantic instant I saw millions of acts both delightful and awful; not one of them occupied the same point in space, without overlapping or transparency. What my eyes beheld was simultaneous, but what I shall now write down will be successive, because language is successive. Nonetheless, I'll try to recollect what I can.

J.L. Borges, *"The Aleph"*

Chapter 2

State of the Art

The chapter is devoted to provide a theoretical background for the rest of the chapters. We introduce the notion of constraint problems and then focus on distributed constraint satisfaction and optimisation problems. Further, we provide an overview of auctions and then go into more detail to explain combinatorial auctions and the algorithms to clear them. Finally, we detail the current status in the deregulation of European energy markets.

2.1 Constraint Problems

Constraint problems constitute one of the most powerful problem solving paradigms in Artificial Intelligence[1] [FY05]. Formally speaking, a constraint problem is defined by a set of variables $V = \{X_1, X_2, \ldots X_n\}$, with each having an associated non-empty

[1] Nowadays, the general banner "Artificial intelligence" is also known as "Computational Intelligence", (CI).

domain of possible values. Moreover, there is a set of constraints $C = \{C_1, C_2,\ C_m\}$ which contain subsets of V and specify a relationship within that subset according to some constraint language. Basically, each constraint C_i specifies the possible values for the subset of variables it involves. In this way, a *state* of the problem is a set of assignments to some or all of the variables. This is, $\{X_i = v_i, X_j = v_j \ldots\}$. States are *consistent* or legal when they do not violate any constraint. In case all variables are mentioned within the state, it is *complete*. And, if a complete state is consistent, then we have a solution.

Depending on the goal of the problem, we can deal with a constraint *optimisation* (COP) or constraint *satisfaction* (CSP) one. In the first type, constraints cannot be violated (thus, they must be satisfied). In the latter, there exist a *fitness function* (aka objective or cost function) that evaluates the state (i.e how good or bad does it perform) and the goal is to obtain the combination of variable assignments that maximise or minimise (depending on the problem) the function. With this approach, constraint satisfaction problems can be seen as a subgroup of constraint optimisation ones since they look for a combination that gets a fitness value of 0 (i.e. where no constraint is violated).

Now, focusing on constraint satisfaction, let us illustrate the definitions above with two archetypal constraint satisfaction problems: the *map colouring* problem and the *n-queens* problem. The formulation of the map colouring problem is as follows: given a certain map, the task consists in finding the combination of colours for each territory where neighbouring territories have a different colour with the minimum possible number of colours. Figure 2.1 shows an example of the map-colouring problem.

In order to ease its visualisation, it is also possible to use a constraint graph where each country is represented as a node and the edges connecting two of them are the constraints. Thus, in the example of Figure 2.1, Austria is surrounded by all the rest whereas, for instance, Switzerland only borders upon Austria. An incomplete state could be $S_1 = \{$A = red, H = white, SL = white$\}$, which will be non consistent as well, since Hungary and Slovenia would have the same colours though being neighbours.

In the N-Queens problem, the challenge consists in, having an n x n draught-board, placing n queens so that no two queens threaten each other. Thus, no two queens can be on the same row, column or diagonal. Usually, the problem is finished after finding a consistent combination for the queens but sometimes it may be asked to list all possible legal solutions. Figure 2.2 shows some consistent and inconsistent situations with four queens.

Since there are as many queens as rows, we can assure that consistent solutions

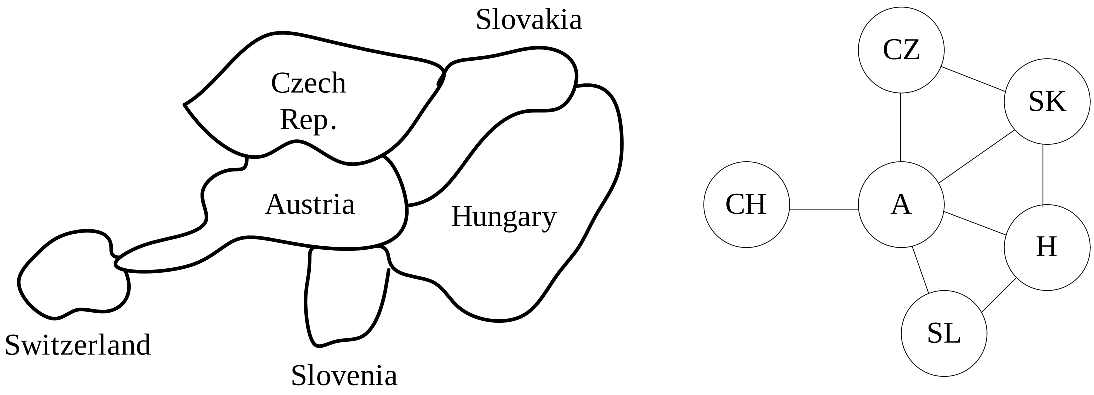


Figure 2.1: **Map Colouring Problem** – The map of central Europe to be coloured (left) and its constraint graph (right) to solve it as a CSP.

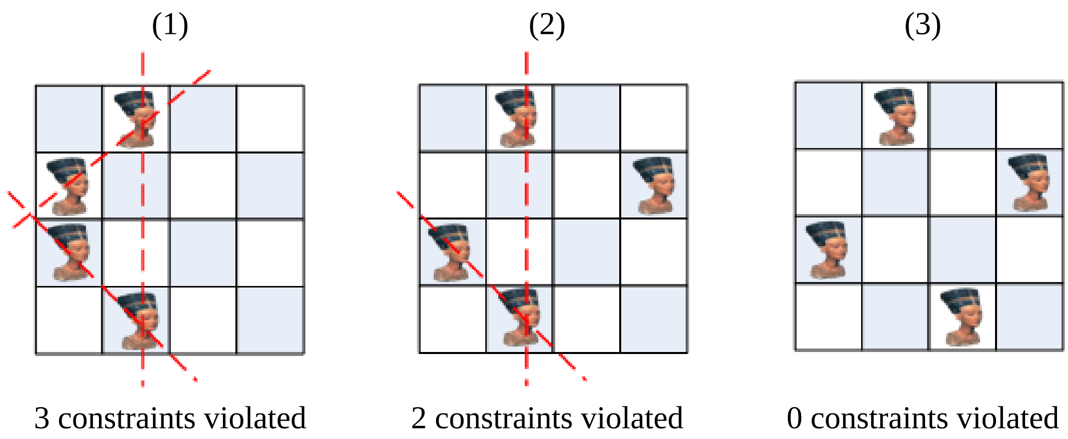


Figure 2.2: **4-Queens Problem Example** – Situations (1) and (2) are inconsistent with respectively 3 and 2 constraints violated, and situation (3) shows a legal combination where no queen threatens any other.

will have every queen placed in a different row. In this way, the problem can be formalised as a CSP with n variables, each of which corresponds to the position of a queen in her row. Moreover, the domain of the variable is $[1..n]$ (e.g. in the 4-Queens problem, *1, 2, 3, 4*) and the constraints can be articulated as binary predicates. For instance, a constraint between q_i and q_j will be represented as $q_i \neq q_j \wedge \mid i - j \mid \neq \mid q_i - q_j \mid$. Finally, a solution is a combination of the values of all variables and the goal is to find a consistent combination of them.

The advantages of modeling a problem as a CSP are many. As stated by Russell and Norvig in [RN02]:

> Because the representation of states in a CSP conforms to a *standard* pattern –that is, a set of variables with assigned values– the successor function and goal test can be written in a generic way that applies to all CSPs. Furthermore, we can develop effective, generic heuristics that require no additional domain-specific expertise. Finally, the structure of the constraint graph can be used to simplify the solution process, in some cases giving an exponential reduction of complexity.

Unfortunately, finding a consistent solution is NP-complete and thus, a trial-and-error exploration of the possible alternatives is inevitable [YH00]. In this way, methods for solving CSPs can be divided into two groups, namely *search* algorithms and *consistency* algorithms [Mac92, YIDK92, YH00], where search algorithms can be further divided into *backtracking* algorithms and *iterative improvement* algorithms.

Let us first go into a deeper detail with backtracking algorithms, which represent a class of basic, systematic search algorithms for solving CSPs. The general banner of backtracking refers to the action executed by the algorithm when there are no legal values left to assign [RN02]. Basically, a backtracking algorithm involves a depth-first search that for each single variable tests all its legal values and then backtracks, as shown in Figure 2.3.

The algorithm starts working with a partial solution consisting in the value assignment of one variable. This partial solution is extended by adding new variables one by one, until it becomes a complete solution. In case that one variable has no legal value that satisfies all of the constraints with the partial solution, the value of the last-added variable is changed in the backtracking process that names the algorithm.

Nevertheless, though being complete, the performance of this algorithm is poor since it systematically tries all values of a variable *without considering which could be better*. To remove this shortcoming, we can use a value-ordering heuristic that somehow favours promising values so a consistent solution can be found faster. In this way, the min-conflict heuristic [MJPL92] chooses the least problematic value each time. It starts with an initial tentative value for each variable, which, when the value is added to the partial solution must satisfy all the previous constraints (as in every backtracking algorithm). If there exist several values that fulfill this condition, the min-conflict heuristics selects the one that satisfies the highest number of constraints with tentative initial values.

function BACKTRACKING-SEARCH(*csp*) **returns** a solution, or failure
return RECURSIVE-BACKTRACKING({ }, *csp*)

function RECURSIVE-BACKTRACKING(*assignment*,*csp*) **returns** a solution, or failure
if *assignment* is complete **then return** *assignment*
var ← SELECT-UNASSIGNED-VARIABLE(VARIABLES[*csp*], *assignment*, *csp*)
for each value **in** ORDER-DOMAIN-VALUES(*var*, *assignment*, *csp*) **do**
if *value* is consistent with *assignment* according to CONSTRAINTS[*csp*] **then**
add {*var* = *value*} to *assignment*
result ← RECURSIVE-BACKTRACKING(*assignment*, *csp*)
if *result* ≠ *failure* **then return** *result*
remove {*var* = *value*} from *assignment*
return *failure*

Figure 2.3: **Backtracking Algorithm for Constraint Satisfaction Problems [RN02] –** The pseudocode illustrates the depth-first search carried out by the algorithm. When a variable has no legal values left to assign then the algorithms backtracks to assign a new value to its predecessor.

```
function ITERATIVE IMPROVEMENT (csp, max steps) returns a solution or
failure
  inputs: csp, a constraint satisfaction problem
    max steps, the number of steps allowed before giving up

   current ← an initial complete assignment for csp
   for i = 1 to max_steps do
     if current is a solution for csp then return current
     var ← a randomly chosen, conflicted variable from VARIABLES[csp]
     value ← the value v for var that minimises CONFLICTS(var, v, current, csp)
    set var = value in current
  return failure
```

Figure 2.4: **Iterative Improvement Algorithm with Min-Conflict Heuristic for Constraint Satisfaction Problems [RN02] –** The pseudocode illustrates the hill-climbing search guided by a min-conflict heuristic.

The other class of search algorithms are those known as iterative improvement algorithms ("local search algorithms" in [RN02]). In this case, the algorithms perform a hill-climbing search on an initial tentative set of values. Therefore, the initial non-consistent solution is iteratively improved. This time as well, the search process can be guided heuristically. For instance, by choosing the variable value that minimises the number of violated constraints (again, min-conflict heuristic). Figure 2.4 shows an iterative improvement algorithm with min-conflict heuristic.

Being hill-climbing algorithms, a major flaw of iterative improvement is that they may occasionally get trapped in *local-minima*. These are states in which no change in the value of a single variable improves the situation (i.e. decreases the number of violated constraints). Therefore, since the algorithm finds no better move, it gets blocked without being able to continue (though the current solution is not consistent). For instance, Figure 2.5 shows a local-minimum situation in the four-queens problem. The draughtboard on the left presents the start situation with 2 constraints violated. The rest of the draughtboards develop this situation by changing the value of one queen.

Now, one of the methods proposed to escape from these situations is the breakout algorithm [Mor93]. Each constraint is assigned an initial weight of 1. When trapped in a local minimum, the algorithm increments the weights of the violated constraints. In this way, it can use the summation of the weights of violated constraints as a fitness

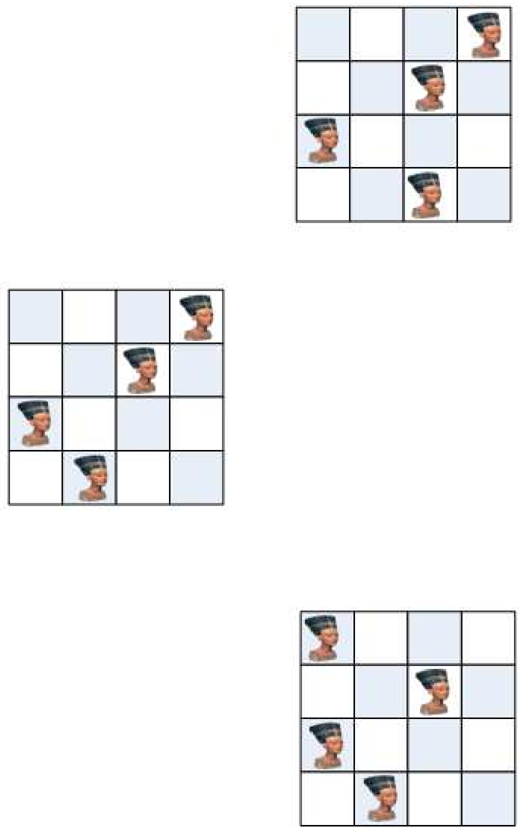


Figure 2.5: **Local-minima in the 4-Queens Problem [RN02] –** Changing the value of only a single variable does not reduce the number of overall violated constraints. In the initial situation (left), this number is 2. The other combinations on the right, obtained after changing only one variable, cannot do better.

function that favours states close to the local minimum and, therefore, helps to escape from it.

Nevertheless, the search may not be exhaustive and therefore, iterative improvements algorithms, though efficient, cannot be guaranteed to be complete [YH00]. Still, they are surprisingly effective for many CSPs, particularly when given a reasonable start situation. For instance, in the n-queens problem, the runtime of the iterative improvements algorithms is roughly independent of the problem size. As [RN02] states, “it solves even the million-queens problem in an average of 50 steps (after the initial assignment)”. The reason for this phenomenon is that “the solutions are densely distributed throughout the state space” [RN02].

Consistency algorithms [Mac92] constitute an alternative to search algorithms. Basically, they are preprocessing algorithms with the purpose of reducing unnecessary backtracking. Following [HY00], the can be classified according to the notion of *k-consistency*:

> A CSP is k-consistent iff given any instantiation of any $k - 1$ variables satisfying all the constraints among those variables, it is possible to find an instantiation of any kth variable such that the k values satisfy all the constraints among them. If there are n variables in a CSP and the CSP is k-consistent for all $k \leq n$, then a solution can be obtained immediately without any backtracking.

The problem is that getting to that high degree of consistency requires prohibitively high computational costs. Therefore, the solution lies in finding the proper combination of consistency algorithms and backtracking.

2.1.1 Distributed Constraint Satisfaction Problems

All the algorithms presented above are centralised. This is, one agent owns all the knowledge about the problems and searches alone for the solution. Nevertheless, in some situations it may be inadequate or even impossible to proceed in this way. Basically, the reasons that prevent from solving a CSP from being solved in a centralised manner are [FY05]:

- *The cost of creating a central authority*: When the problem is per se distributed among a set of peer agents, having a central authority to solve the problem implies adding another element not present in the original architecture.

- *The cost of transferring the knowledge*: In some occasions, constraints or preferences are difficult to express or articulate. Transferring this knowledge to a central agent would require detailing the articulation of the constraint for all possible situations and thus, would be prohibitively costly.
- *Security/privacy concerns [man01]*: The preferences or constraints may be strategic information, unable to be revealed to competitors or even to a central authority. This is often the situation in e-commerce.
- *Having a central point of failure*: If the problem is solved in a distributed manner and one agent fails, the rest may still be able to find a solution without it, as happens in sensor networks. On the contrary, the failure of the central authority may be critical.

Moreover, in many cases, the process may be speeded up by being carried out in parallel by many agents. Together all these reasons point to solving the problem in a distributed fashion[2], this is, as a Distributed Constraint Satisfaction Problem (dCSP[3]).

Makoto Yokoo has helped pioneer the research on this area. Roughly, a dCSP is a CSP in which variables and constraints (i.e. the knowledge of the problem) are distributed among many agents and these agents are required to satisfy all constraints by communicating with each other [HY97]. Therefore, we can say that the goal now is to find consistent combinations of actions that lead to the satisfaction of the inter-agent constraints [YH00].

Distributed Constraint Satisfaction problems can be defined formally as follows [YH00]:

> There exist m agents 1, 2, $\ldots m$. Each variable x_j belongs to one agent i (this relation is represented as $belongs(x_j, i)$). Constraints are also distributed among agents. The fact that an agent l knows a constraint predicate p_k is represented as $known(p_k, l)$. We say that a Distributed CSP is solved iff the following conditions are satisfied.
>
> - $\forall i, \forall x_j$ where $belongs(x_j, i)$, the value of x_j is assigned to d_j, and

[2] Parallel or distributed processing methods for solving CSPs are not really suited for distributed CSPs [CDK91, ZM91] since first, they are designed for efficiency, and second, the reasons that prevent us from using a central authority apply for them as well [YH00].

[3] Often in the literature also written as ”disCSP” or simply ”distributed CSP”.

- $\forall l, \forall p_k$ where $known(p_k, l)$, p_k is true under the assignment $x_j = d_j$.

Various application problems in MAS can be modeled as a distributed CSP. For instance, the distributed interpretation problem [LC88], the distributed resource allocation problem [CKL91], the problem in multi-agent truth maintenance tasks [HB91], and the distributed scheduling problem [SRSKF91], which presents some similarities with the scheduling problem we face in this thesis.

As a general rule, all these examples assume the following communication framework [YDIK98]:

- The communication between agents is achieved by exchanging messages but the address of the receiver must be known in beforehand.
- The communication is reliable and thus, messages are received in the same order in which they are sent.

A more exhaustive overview of the research in distributed constraint satisfaction problems until 2000 can be found in [Yok01]. Now, let us introduce two classical dCSP algorithms that will be used in Chapter 5, *Asynchronous Backtracking* [YIDK92, YDIK98, YH00] and *Asynchronous Weak-Commitment* [YDIK98, YH00], since they can be adapted to work within our problem domain. In the first one, Asynchronous Backtracking, the dCSP is solved by an asynchronous exchange of messages. Agents form a chain ordered by priority and each agent is responsible for enforcing all constraints between itself and all variables owned by higher agents in that chain. In this way, each agent sends concurrently the value of its variable to the next agent in the chain within an *ok?* message, asking whether the value is acceptable. If not, it receives a *nogood* notification, meaning that the other agent has found a constraint violation. Each agent maintains a memory of what other agent's values are (the so-called *agent view*, the values arrived within *ok?* messages). Thus, they can compare this agent view to all received *nogoods* and to their current value sent within the *ok?* messages to find a consistent (non-conflicting) value. In case it is not possible, the assignments of the other agents must be changed to get to a combination that does not violate any constraint and therefore, the agent causes a *backtrack* process and the neighbours start searching for a non-conflicting value. Figure 2.6 shows the pseudocode corresponding to this algorithm.

The asynchronous backtracking algorithm either finds a solution that satisfies all constraints and, if there are none, it terminates. Retaining all nogoods assures

ASYNCHRONOUS BACKTRACKING ALGORITHM

when received (**ok?**, (x_j , d_j)) **do –**
 add (x_j , d_j) to *agent view*;
 check_agent_view;
end do;

when received (**nogood**, x_j , *nogood*) **do –**
 add *nogood* to *nogood_list*;
 when (x_k,d_k) where x_k is not connected is contained in *nogood* do
  request x_k to add a link from x_k to x_i;
  add (x_k, d_k) to *agent view*; **end do;**
 old value ← *current value*; **check_agent_view**; **–**
 when *old_value* = *current_value* **do**
  send (**ok?**, (x_j , *current_value*)) to x_j ; **end do; end do**;

procedure **check_agent_view**
 when *agent_view* and *current_value* are not consistent **do**
  if no value in D_i is consistent with *agent_view* **then backtrack**;
  else select *d* ∈ D_i where *agent_view* and *d* are consistent;
   current value ← *d*;
   send (**ok?**, (x_i, d)) to outgoing links; **end if; end do**;

procedure **backtrack**
 nogoods ← {V | V = inconsistent subset of *agent_view*}; **–**
 when an empty set is an element of *nogoods* **do**
  broadcast to other agents that there is no solution,
   terminate this algorithm; **end do**;
 for each V ∈ *nogoods* **do**;
  select (x_j; d_j) where x_j has the lowest priority in V; **–**
  send (**nogood**, x_i, V) to x_j ;
 remove (x_j; d_j) from *agent_view*; **end do**;
 check_agent_view;
 end do;

Figure 2.6: **Asynchronous Backtracking Algorithm for Distributed Constraint Satisfaction Problems [YDIK98] –** Procedures for receiving messages and performing a backtrack.

the completeness of the asynchronous backtracking algorithm. Nevertheless, high priority agents tend to have a strong commitment to its variable value [HY00]. This is, a high priority agent selecting a wrong value to its variable forces lower priority agents to perform an exhaustive search to revise that wrong value.

This flaw has been tackled in the asynchronous weak-commitment algorithm [Yok95], the second of the classical dCSP algorithms, we present here. Basically, an asynchronous weak-commitment algorithm is an enhancement of the asynchronous backtracking, where agents change their priority values dynamically in order to avoid the strong commitment of high priority agents, mentioned above [HY00]. Moreover, this algorithm uses $ok?$ and $nogood$ messages in a similar way as asynchronous backtracking and it also creates and sends a nogood when the agent finds no consistent value to its variable. Yet then, it increases its priority value to the maximum among its neighbours, which is what allows lower priority agents to revise the wrong values of higher priority ones. The pseudocode corresponding to the Asynchronous Weak-Commitment Algorithm is shown in Figure 2.7.

Weak-commitment algorithms are more efficient than their asynchronous backtracking counterparts and, by recording all nogoods, they are also complete [HY00].

2.1.2 Distributed Constraint Optimisation Problems

Constraint satisfaction has been successfully applied to a wide range of problems, as stated before. Nevertheless, some situations in real life are over-constrained [CDK91, CKLM92, DTWZ94]. This is, no solution satisfies all constraints completely [Yok93, HY97, HY00], so the problem consists in finding a solution that improves or optimises the current one.

In this way, distributed constraint optimisation problems (dCOP) [YD91] can be seen as a generalisation of distributed constraint satisfaction problems, similarly to their centralised versions. The difference lies in the result of the objective function. Whereas in constraint satisfaction it issues the number of constraints violated (so the goal is to reduce it to 0), in constraint optimisation it evaluates the fitness of a set of values of the agents' variables. For instance, in the N-Queens problem, we can say whether a set of assignment constitutes a solution or, if all the space solution has been unsuccessfully processed, that no solution can be found (as for the 3-Queens problem). Think now of a scheduling problem, akin to the one presented in the next chapters. For a given energy demand, we are asked to find the cheapest consumption schedule according to a simple tariff (each time slot having a price for the kWh consumption). Every different set of assignments will have a fitness,

ASYNCHRONOUS WEAK-COMMITMENT ALGORITHM

when received (ok?, (x_j , d_j, *priority*)) **do –**
add (x_j , d_j, *priority*) to *agent view*;
check_agent_view;
end do;

when received (**nogood**, x_j , *nogood*) **do –**
add *nogood* to *nogood_list*;
when (x_k,d_k,*priority*) where x_k is not in *neighbours*
is contained in *nogood* **do**
add x_k to *neighbours*, add (x_k,d_k,*priority*) to *agent_view*; **end do**;
check_agent_view;
end do;

procedure **check_agent_view**
when *agent_view* and *current_value* are not consistent **do**
if no value in D_i is consistent with *agent_view* **then backtrack**;
else select $d \in D_i$ where *agent_view* and *d* are consistent
and *d* minimises the number of constraint violations
with lower priority agents; **–**
current value ← *d*;
send (**ok?**, (x_i, d, *current priority*)) to *neighbours*; **end if**; **end do**;

procedure **backtrack –**
nogoods ← $\{V \mid V$ inconsistent subset of *agent_view*$\}$;
when an empty set is an element of *nogoods* **do**
broadcast to other agents that there is no solution,
terminate this algorithm; end do;
when no element of *nogoods* is included in *nogood_sent* **do**
for each V ∈ *nogoods* **do**;
add V to *nogood_sent*
for each $(x_j; d_j; p_j)$ in V **do**;
send (**nogood**, x_i, V) to x_j ; **end do; end do**;
$p_{max} \leftarrow max(x_j; d_j; p_j) \in agent_view(p_j)$);
current_priority ← p_max;
select d ∈ D_i where d minimises the number of constraint violations
with lower priority agents;
current_value ← d;
send (**ok?**, (x_i, d, *current_priority*)) to *neighbours*; **end do**;

Figure 2.7: **Asynchronous Weak-Commitment Algorithm for Distributed Constraint Satisfaction Problems [YDIK98] –** Procedures for receiving messages and performing a backtrack.

consisting simply of the sum of each time-slot's consumption multiplied by the cost of energy in that time-slot. Therefore, each possible solution will do *better* or *worse*, but we will not be able to select any of then as a consistent one, as in constraint satisfaction. In theory, it is necessary to process all possible solutions to find the best one [4].

The formalisation of constraint optimisation problems is as follows [ML04]:

- a set of n variables V = $\{x_1, \ldots, x_n\}$
- discrete, finite domains for each of the variables D = $\{D_1, \ldots, D_n\}$
- a set of cost functions $f = \{f_1, \ldots, f_m\}$ where each $f_i(d_{i,1}, \ldots, d_{i,j})$ is function $f_i : D_i \times \ldots \times D_{i,j} \rightarrow N \cup \infty$

The problem is to find an assignment $A^* = \{d_{i,1}, \ldots, d_n | d_i \in D_i\}$ such that the global cost is minimised.

As in constraint satisfaction, if the knowledge of the constraint optimisation problem is distributed among several agents, then we have a distributed constraint satisfaction problem (dCOP) [HY97, HY00, LS95, PWF$^+$97]. The formal definition of a dCOP is as follows [MSTM05, PF05b, PF05a]:

- a set of n variables V = $\{x_1, \ldots, x_n\}$, each one assigned to an agent
- discrete, finite domains for each of the variables D = $\{D_1, \ldots, D_n\}$. Only the agent who is assigned a variable has control of its value and knowledge of its domain
- a set of cost functions $f = \{f_1, \ldots, f_m\}$ where each $f_i(d_{i,1}, \ldots, d_{i,j})$ is function $f_i : D_i \times \ldots \times D_{i,j} \rightarrow N \cup \infty$

Each agent chooses its values such that a given global objective function is minimised. The objective function is described as the summation over a set of cost functions, and they are the analogue of constraints from DisCSP (and, thus, referred hereafter simply as "constraints").

---

[4] Applying either a proper heuristic or knowledge of the problem could help avoid this phenomenon. For instance, if we knew that off-peak time-slots are cheaper than the peak ones, we could assure that a solution having all its consumption placed off-peak would do better than another only consuming at the peak times.

The problem is to find an assignment $A^* = \{d_{i,1}, \ldots, d_n | d_i \in D_i\}$ such that the global cost is minimised.

The dCOP framework can be used as coordination mechanism of the decision-making of multi-agent system. For instance, [MSTM05] lists:

> Satellite constellations [Bar99], disaster rescue [KTN$^+$99], multi-agent teamwork [Tam97], human/agent organizations [CGK$^+$01], intelligent forces [CSC$^+$93], distributed and reconfigurable robots [SY02] and sensor networks [Sys01] as some examples of multi-agent applications where distributed reasoning problems arise.

One may think that, since dCOP is an extension of the dCSP framework, dCSP algorithms could be adapted to optimisation and still work optimally. This is, unfortunately, not possible, as claimed in [MSTM05]:

> Simple extensions to Asynchronous Backtracking for optimisation have relied on converting an optimisation problem into a sequence of dCSPs using iterative thresholding [HY00]. This approach has applied only to limited types of optimisation problems (e.g. Hierarchical dCSPs, Maximal dCSPs), but has failed to apply to more general dCOP problems, even rather natural ones such as minimising the total number of constraint violations (MaxCSP).

That is, dCSP are not successful while *exported* to a dCOP domain. Therefore, we need to use algorithms tailored to dCOP's features. In this way, the Synchronous Branch and Bound (SynchBB [HY97]), is the application of the branch and bound method [Fre89, FW92] to dCOP. Thus, supposing that the solution space can be represented as a tree, if the algorithm knows that the optimal solution cannot occur in any of the successors of a certain node, there is no need to continue exploring solutions that include successors from that node[5]. And this is roughly the philosophy of SynchBB [HY97]. The agents interchange sequentially extended partial paths with their value and abandon those whose evaluation is less than the current upper bound. If values are exhausted, the agent backtracks to the previous agent. Figures 2.8 and 2.9 show more details on the SynchBB algorithm.

[5] For instance, looking for the shortest route connecting Vienna and Berlin we may come across a partial route from Vienna to Paris (i.e. the final route would be Vienna-Paris-Berlin). Supposing that the best route found so far requires 700 km and that this Vienna-Paris one already needs 1300 km, we can abandon all routes starting with Vienna-Paris (e.g. Vienna-Paris-Munich-Berlin, Vienna-Paris-Frankfurt-Berlin, and so on) because they will be for sure worse than the current best one.

SYNCHRONOUS BRANCH AND BOUND ALGORITHM (1)

procedure **initiate**
  $d_i \leftarrow$ first value in $domain$;
  send (**token**, [[$x_i, d_i, 0$]], n_i) to the next agent;

when i received (**token**, *current_path*, ub) from the previous agent **do**
  previous_path $\leftarrow$ *current_path*;
  $n_i \leftarrow ub$;
  next $\leftarrow$ **get_next**(*domain*);
  send_token; end do;

when i received (**token**, *current_path*, *ub*) from the next agent **do**
  [x_i, d_i, nv_i] $\leftarrow$ the element related to x_i in *current_path*;
  $n_i \leftarrow ub$;
  next $\leftarrow$ **get_next**(*domain* minus all elements up to d_i)
  send_token; end do;

procedure **send_token**
  if *next* neq exhausted **then**
    if i = the last agent **then**
      next_to_next $\leftarrow$ *next*;
      while *next_to_next* $\neq$ exhausted **do**
        best_path $\leftarrow$ *new_path*;
        $ni \leftarrow$ max nv_j in *best_path*;
        when $n_i \leq s_i$ **do**
          terminate the algorithm; **end do**;
        next_to_next $\leftarrow$ get_next(domain minus all elements up to *next_to_next*);
      end do;
  send (**token**, *previous_path*, n_i) to the previous agent;
    else
      send (**token**, new_path, n_i) to the next agent; **end if**;
  else
    if i = the first agent **then**
      terminate the algorithm;
    else
      send (**token**, *previous_path*, ni) to the previous agent; **end if; end if**;

Figure 2.8: **Synchronous Branch and Bound Algorithm for Distributed Constraint Optimisation Problems [HY97] (1) –** *initiate* and *send_token* procedures carried out by the agents.

SYNCHRONOUS BRANCH AND BOUND ALGORITHM (2)

procedure **get_next**(*domain*)
  if *domain* = nil **then**
    return exhausted;
  else
    $d_i \leftarrow$ first value in *domain*;
    new_path $\leftarrow$ nil;
    counter $\leftarrow$ 0;
    if check(*previous_path*) **then**
      return d_i;
    else
      return get_next(*domain* minus d_i); **end if; end if**;

procedure **check**(*path*)
  if *path* = nil **then**
    append [x_i, d_i, *counter*] to *new_path*;
  return *true*;
  else
    $[x_j, d_j, nv_j]$ $\leftarrow$ first element in *path*;
    if $[x_i, d_i]$ and $[x_j, d_j]$ are not consistent **then**
      counter $\leftarrow$ *counter* + 1;
      if *counter* leq n_i or nv_j + 1 $\leq n_i$ **then**
        return *false*
      else
        append [xj, dj, nvj + 1] to *new_path*;
        return check(*path* minus first element); **end if**;
    else
      append $[x_j, d_j, nv_j]$ to *new_path*;
      return check(*path* minus first element); **end if; end if;**

Figure 2.9: **Synchronous Branch and Bound Algorithm for Distributed Constraint Optimisation Problems [HY97] (2) –** *get_next* and *check* procedures carried out by the agents.

SynchBB performs an exhaustive search and thus, is complete [HY97, MSTY03] (i.e. if it exists, it finds an optimal solution and otherwise, it terminates). The major flaw of SynchBB lies precisely on its synchrony. It does not allow agents to change their values in parallel. Therefore, according to [MSTM05] it is prohibitively slow due to its synchronous, sequential communication.

To remove this shortcoming, a new algorithm called Asynchronous Distributed OPTimisation (Adopt) [MSTY03, Mod03, MSTM05] replaces the depth-first search strategy with an asynchronous best-first search and the upper bound with a lower bound as comparison mean for abandoning less promising solutions. This is one of the special features of Adopt: it may abandon partial solutions before suboptimality is proved and, therefore, it requires a mechanism to efficiently reconstruct some of the previously dismissed solutions (called *backtrack threshold*). Indeed, Adopt is a backtracking search that maintains a lower and an upper bound at each variable during its search. It orders the nodes in a tree where constraints join ancestors and descendants and no constraints are allowed among siblings [MSTY03] (i.e. the constraint graph is transformed into a depth-first search tree). The lower and upper bound become progressively closer and the algorithm terminates when it returns to the root agent. Figures 2.11 and 2.12 detail the procedures for receiving messages and for updating backtrack thresholds in Adopt.

Now, Adopt significantly outperforms SynchBB, as shown in Figure 2.10 for the graph colouring problem, using the synchronous cycle as comparison dimension (though, as noted by Modi et al. in [MSTM05], more sophisticated metrics for comparing algorithms are needed[6].

Nevertheless, Adopt, and backtracking algorithms in general, have received lately some critics on their suitability for distributed systems [PF05a]:

> They may not be the best basis since in backtrack search, control shifts rapidly between different variables. Every state change in a distributed backtrack algorithm requires at least one message; in the worst case, even in a parallel algorithm there will be exponentially many state changes [Kas86], thus resulting in exponentially many messages. So far, this has been a serious drawback for the application of distributed algorithms in the real world, especially for optimisation problems (also noted in [MTB$^+$04]).

[6] In the comparison of Chapter 5 among Adopt and our novel dCOP algorithm we use different metrics to this extent, such as the frequency of calls to the cost function and the overall number of messages exchanged.

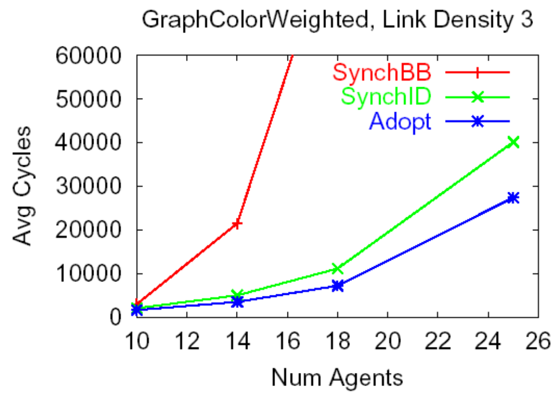


Figure 2.10: **Comparison among Adopt and SynchBB [MSTM05]** – Adopt clearly outperforms SynchBB and the improved version of this, SynchID in the Graph Colouring problem.

As we shown in chapter 5, Adopt and the rest of the existing dCOP and dCSP algorithms are not able to work under the conditions that the problem domain addressed in this dissertation poses. Therefore, there are two alternatives: either adapting those algorithms or developing a novel one. Chapter 5 details how both of these alternatives were carried out.

2.2 Auctions

The etymology of the word auction traces its roots back to the Latin voice *aucti*, *auctin-*, from *auctus*, past participle of *augere*, ”*to increase*”. This meaning is closely associated to the concept that an individual of the western culture recalls when thinking of an auction: the sale of a good in which one or more participants (usually round-wise) offer a buy price increasingly until no one dares to outbid the last one. Formally, an auction is indeed a game of incomplete information [Kri02]; a negotiation protocol[7] where one or more bidders, in a dynamic process, agree with the auctioneer to buy one or more goods or services at a certain price [Wur01]. The key idea is that this price is unknown or variable beforehand. The objective of the auction

[7] [HJL03] mentions another two types of negotiation protocols: *multilateral negotiation*, where the negotiation involves bargaining between multiple noncooperative parties [ARS96] and *n-bilateral* negotiations, where the negotiation process comprises multiple bilateral bargaining encounters [Far00].

ASYNCHRONOUS DISTRIBUTED OPTIMISATION (ADOPT) (1)

initialize
  $threshold \leftarrow 0$; *CurrentContext* $\leftarrow \{\}$
  forall d $\in D_i, x_l \in$ *Children* **do**
    $lb(d, x_l) \leftarrow 0; t(d, x_l) \leftarrow 0$
    $ub(d, x_l) \leftarrow Inf; context(d, x_l) \leftarrow \{\}$; **enddo**
  $d_i \leftarrow d$ that minimises **LB**(d)
  backtrack

when received (**THRESHOLD**, t, *context*)
  if *context* compatible with *CurrentContext*:
    $threshold \leftarrow t$
    maintainThresholdInvariant
    backtrack; **endif**

when received (**TERMINATE**, *context*)
  record TERMINATE received from parent
  CurrentContext $\leftarrow$ *context*
  backtrack

when received (**VALUE**, (x_j, d_j))
  if TERMINATE not received from parent:
    add (x_j, d_j) to *CurrentContext*
    forall $d \in D_i; x_l \in$ *Children* **do**
      if *context*($d; x_l$) incompatible with *CurrentContext*:
        $lb(d, x_l) \leftarrow 0; t(d, x_l) \leftarrow 0$
        $ub(d, x_l) \leftarrow Inf$; *context*$(d, x_l) \leftarrow \{\}$; **endif**; **enddo**
    maintainThresholdInvariant
    backtrack; **endif**

Figure 2.11: **Asynchronous Distributed OPTimisation Algorithm (Adopt) for Distributed Constraint Optimisation Problems [MSTM05] (1) –** Procedures for receiving messages.

ASYNCHRONOUS DISTRIBUTED OPTIMISATION (ADOPT) (2)

when received (**COST**, x_k, *context*, lb, ub)
  $d \leftarrow$ value of x_i in *context*
  remove (x_i, d) from *context*
  if TERMINATE not received from parent:
    forall $(x_j, d_j) \in$ *context* and x_j is not my neighbour **do**
      add (x_j, d_j) to *CurrentContext*;**enddo**
    forall $d' \in D_i, x_l \in$ *Children* **do**
      if *context*(d0; xl) incompatible with *CurrentContext*:
        $lb(d', x_l) \leftarrow 0; t(d', x_l) \leftarrow 0$
        $ub(d', x_l) \leftarrow Inf$; *context*$(d', x_l) \leftarrow \{\}$;**endif**;**enddo**;**endif**
  if *context* compatible with *CurrentContext*:
    $lb(d, x_k) \leftarrow lb$
    $ub(d, x_k) \leftarrow ub$
    context(d, x_k) *context*
    maintainChildThresholdInvariant
    maintainThresholdInvariant; **endif**
  backtrack

procedure backtrack
  if $threshold$ == **UB**:
    $d_i \leftarrow d$ that minimises **UB**(d)
  else if LB$(d_i) > threshold$:
    $d_i \leftarrow d$ that minimises **LB**(d);**endif**
  SEND (**VALUE**, (x_i, d_i)) to each lower priority neighbour
  maintainAllocationInvariant
  if $threshold$ == **UB**:
    if TERMINATE received from parent or x_i is root:
      SEND (**TERMINATE**, *CurrentContext* $\cup \{((x_i, d_i)\}$) to each child
      Terminate execution; **endif**;**endif**
  SEND (**COST**, x_i, *CurrentContext*, **LB, UB**) to parent

Figure 2.12: **Asynchronous Distributed OPTimisation Algorithm (Adopt) for Distributed Constraint Optimisation Problems [MSTM05] (2) –** Procedures for receiving messages.

is actually to settle this price. In case of a *normal* sale process[8], clients just consider whether buying or not (e.g. "is this item worth this price?") and buy it if the price is less than their own valuation. For an auctioneer, an auction represents an opportunity of increasing the revenue from a fixed-price sale in the same way that for the bidders, an auction represents an opportunity to obtain the good for a lower amount than their valuation. In other words, an auctions enables them to gain something. Nevertheless, if they pay an amount equally to their valuation, they obtain no benefit, since, at the end of the day, they are indifferent between the money and the good.

Two issues arise at this point: *bid shading* and the phenomenon termed as *winner's curse*. Let us start with the latter. The winner's curse is a logical consequence of supposing that all participants are roughly equal able of estimating the value of a good and bidding accordingly. According to this principle, in auctions with incomplete information, where bidders don't know the valuation of the others, the winner has probably overestimated the commodity's valuation since all the rest have placed lower bids (thus, their valuation was less) so the winner may think he has overpaid.

In order to prevent the winner's curse, sensible bidders may place a bid below their valuation (i.e. they do bid shading) to be sure they don't pay more than the good is worth. The use of bid shading implies the possibility of using strategies to pay less than their valuation. We will discuss afterwards how the presence of incomplete information boosts strategies, and how can the auctioneer prevent it to assure the maximum pay-off.

Now, we have discussed so far what an auction *is* but not *what it looks like*. An auction may actually present many varied forms. For instance, they may differ in the number of distinct items to be auctioned, in whether there exists a reserve price and whether it is known to bidders[9], in the way bids are collected, in the mechanism for winner determination and the price the winner pays, in the participants' allowed bids or in how often is it possible to bid and the way bids are presented. As can be seen, there are many aspects to take into account. Table 2.1 summarises seven main dimensions of this kind that will allow us to describe and compare several well-known classical auction paradigms.

Common auction type lists usually cite the four most popular auction formats, all single-sided[10] auctions: first-price ascending (*English*), first-price sealed-bid, first-

[8] A *normal* sale process is, for instance, to do the shopping in a supermarket. In opposition to any kind of negotiation protocol, the prices are fixed by the seller and there is no space left for bargaining.

[9] Another aspect to take into account is the type of information that can be revealed in the process [Fri93]. [WWW98] gives an accurate discussion on how can this phenomenon influence the outcome.

[10] Single-sided auctions are those where buyers submit asks and bidders place bids. On the contrary,

Dimension	**Values**	**Value explanation**
Auction mode	one-sided (O) two-sided (T)	only bids or asks are permitted both bids and asks are permitted
Duration time	single-round (S) multi-round (M)	the auction lasts one round the auction lasts multiple round
Number of goods	one (O) many (M)	only one good auctioned multiple goods are auctioned
Buyer:Seller Ratio	many to one (n:1) one to many (1:m) many to many (n:m)	multiple buyers and only one seller multiple sellers and only one buyer multiple buyers and multiple sellers
Information Revealed	yes no	intermediate information revealed bidder has no information about others
Settlement Price	first price (F) second price (S) different prices (D)	highest price among all the bidders second highest price among all the bidders prices vary for each trade
Closing rules	time (T) inactivity (I) budget (B)	when a time point is reached when there are no more bids for a time period when a buy-out price is reached

Table 2.1: **Main dimensions for a comparison among different auction types. –** Based in [HJL03], the dimensions will be used in the comparison of table 2.2.

price descending (*Dutch*), and second-price sealed-bid (*Vickrey*) [San99, Vic61]. The fifth item in the following account is the most common type of double-sided auction: the continuous double auction. All these formats auction a single nondivisible good.

- *English Auction*: This is what most people think of when speaking about an auction. It is the oldest and probably most prevalent auction form [Kri02]. Bidders place their bids openly, each bid being always higher than the last until no more bids are placed. Alternatively, some English auction forms include a *buy-out price* which, when reached, automatically terminates the auction. Finally, the seller may impose a *reserve price* (public or not) under which the commodity cannot be sold. The participant that places the highest (i.e. last) bid wins the auction and pays the price offered in this last bid.
- *Dutch Auction*[11]: In this auction, the auctioneer starts with high a price, which is decreased gradually until somebody is willing to pay the announced amount. This auction format is useful when the auctioneer wants to sell the goods quickly, since only one bid can be placed (the bid accepting the price proposed by the auctioneer) [HJL03].
- *First-Price Sealed-Bid*: Unlike the previous two, this auction is sealed. Therefore, the participants don't know the amount bid by the rest. They just submit their bids and the good goes to the highest one.
- *Vickrey Auction*: Named after the 1996 Nobel Prize in Economic Sciences William Vickrey, whose work was seminal in this area and classified this auction type in the 1960s. Vickrey auctions are like English auctions with just one small peculiarity[12]: the price paid by the winner is not the amount offered in the last and winning bid but in the second last (i.e. second highest bid) [Vic61].
- *Continuous Double Auction (CDA)*: In this auction format, buyers and sellers are allowed to continuously update their bids or asks at any time in the trading process [HJL03].

Now, excepting in the case of the CDA, the auction formats presented so far are embedded in a classical market: one seller and more (potential) buyers. Nevertheless,

in double-sided auctions, buyers and bidders can both submit asks or bids in the same marketplace [HJL03].

[11] The term was coined after its best known example, the Dutch tulip auction. Paradoxically, Dutch people call it "Chinese Auction".

[12] Small peculiarity that makes a big difference in terms of equilibrium implementation, as we will see in section 2.2.1

there exist some other organisations offering interesting features. In this way, from the point of view of the number of buyers and sellers, we can divide auction markets into three types, as illustrated in Figure 2.13:

- *Forward Auctions*: This model is the most popular, consisting of one seller, the auctioneer, and a number of potential buyers. English, Dutch and the most popular auctions presented in table 2.2 (excepting the CDA) are forward auctions.
- *Reverse Auctions*: In opposition to forward auctions, in reverse auction markets, the buyer starts the process, typically by placing an *ask*. Then, buyers offer their services or goods. Reverse auctions are used typically in procurement auctions. The auctions used in the novel market format described in this dissertation are reverse.
- *Exchanges*: Sometimes, there are marketplaces in which many buyers and sellers coexists simultaneously and, moreover, some of them are both buyers and sellers (even in the same bid). One bid of this kind could be, for instance: “I want to buy a Ferrari F40, sell my Ford Fiesta and I’m also ready to pay 10.000 €” [EKLL01, SF02, Sil02]. In case combinatorial bidding is allowed (see section 2.2.2), it is then known as a *combinatorial exchange* [San00a, SS00a]. Formally, forward combinatorial auctions and reverse combinatorial auctions are special cases of combinatorial exchanges, as shown in [SSGL02].

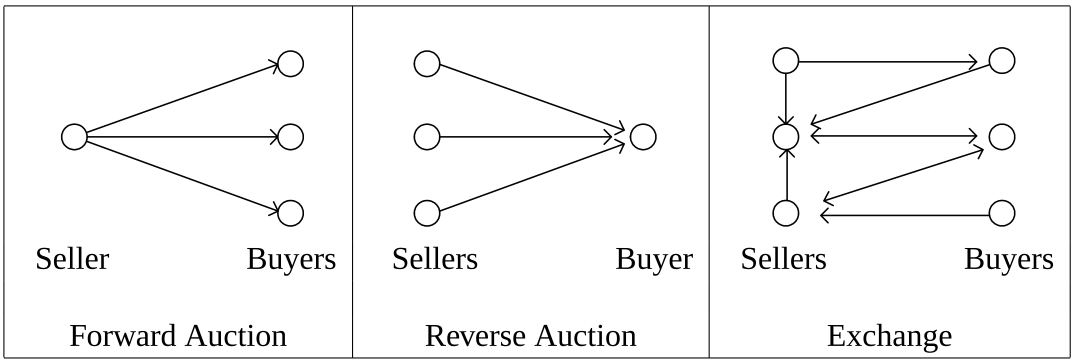


Figure 2.13: **Different Auction Markets –** Depending on the number of buyers and sellers, auction markets can be divided into forward auctions (left), reverse auctions (middle), and exchanges (right).

		English	**Dutch**	**FPSB**	**Vickrey**	**CDA**
Auction Mode	O	√	√	√	√	
	T					√
Duration Time	S			√	√	√
	M	√	√			
Number of goods	O	√	√	√	√	
	M					√
Buyer:Seller Ratio	n:1	√	√	√	√	
	1:m	√	√	√	√	
	n:m					√
Information Revealed	Y	√	√			√
	N			√	√	
Settlement Price	F	√	√	√		
	S				√	
	D					√
Closing Rules	T			√	√	
	I	√				√
	B		√			

Table 2.2: **Comparison of the most popular Auctions –** Based in [HJL03]. Table 2.1 details the dimensions along which the comparison has been done.

Further, we show in table 2.2 a comparison among these five popular auction formats along the dimensions detailed in table 2.1.

As we see, auctions can differ a lot from one another. In fact, Wurman et al. [WWW98] define a taxonomy of parameters that covers almost 25 million different formats of auctions. For instance, the Internet Auction List[13] contains almost 2600 (2005) auction companies. All these facts convert auctions in one of the most studied mechanisms in e-commerce [PUF99]. Moreover, beyond the dimensions of table 2.1, the auctioneer can design its own auction, in order to assure the consecution of its goal[14] by *forcing* to behave in a certain way. And this is exactly the challenge tackled by the so-called *mechanism design problem*. This topic arises when considering an auction as a game and, therefore, part of the game theory, as detailed in the following section.

2.2.1 Auctions as a Game

We have said before that auctions are a negotiation protocol and a game of incomplete information. What do we mean by this statement? *Game Theory*, in comparison to the other microeconomic theory, the *Theory of Competitive Equilibrium*[15], studies and models situations in which decision-makers interact and take their decisions *strategically* (i.e. take into account their knowledge or expectations of other participants' behaviour) [Kre90, OR94, HHV95, DPJ03].

In other words, game-theoretic participants analyse others' behaviour by introspection and deduction, use this information while taking their own decisions and model the effect their actions will have on other agents actions[16] [FT91, Par01]. Therefore, auctions are governed by two interrelated aspects: a general lack of information (hence the banner "game of incomplete information"), and the private interest

[13] http://www.internetauctionlist.com

[14] An auction can be designed either to be efficient or to obtain the highest revenue. From the social point of view, efficiency may be a more desirable goal. In this case, the good or service goes to the participant that values it the most ex-post, and some times, this is not the highest bidder. For a discussion among revenue and efficiency, see [Kri02].

[15] Participants of the situations studied by the Theory of Competitive Equilibrium [Wel93, Cle96], are not interested in their competitors' behaviour or valuations but only in *environmental* parameters, such as the price [DPJ03, OR94, Kre90]

[16] This kind of vicious circle is the classical explanation of Microeconomics Theory to the emergence of equilibrium. For an alternative to this traditional view, see [FL98], where Fudenberg and Levine propose that "equilibrium arises as the long-run outcome of a process in which less than fully rational players grope for optimality over time" (Publisher's note).

of the participants. Otherwise, if the exact and true good's valuation of participants were of public knowledge, it wouldn't be necessary to start the auction at all. The auctioneer could directly sell the good to the participant with the highest valuation.

Moreover, there are two kinds of conflicting interests: among the bidders and among the bidders and the auctioneer. The first is obvious: all participants *compete* for the good, service or commodity. The second refers to the fact that the auctioneer will try to achieve its objective despite the self-interest of individual bidders, and that is exactly the task with which the so-called *mechanism design problem* deals [MCWG95]. As Parkes states [Par01]:

> In words, a mechanism defines the strategies available (e.g., bid at least the ask price) and the method used to select the final outcome based on agent strategies (e.g., the price increases until only one agent bids, then the item is sold to that agent for its bid price).

Mas-Colell et al. put special emphasis on the eventual privacy of some information [MCWG95]:

> An important feature of many settings in which collective decisions must be made is that individuals' actual preferences are not publicly observable. As a result, in one way or another, individuals must be relied upon to reveal this information.

Transposing the subject into the multi-agent system paradigm, the ideal situations happens when the designer of the system can regulate both protocol and strategy for each singular agent, as in [BJW02, PBS03, JB03, PS04, BJW04]. In some situations, however, agents represent different stakeholders and eventually compete against each other. Hence, agents get compelled to follow the protocol, but the designer cannot prevent the adoption of strategies [DPJ03].

Now, before defining a mechanism formally, let us start with its components[17].

Definition 1 (type) *The* type Θ_i *of an agent* i *is a set of pairs* $< o, u_i >$ *where* u_i *expresses agent* i*'s valuation for outcome* o *of a game.*

Therefore, the type of an agent is the description of how it values each outcome of the game. In this way, let $\mathcal{O}$ be the set of possible outcomes. We can then express

[17] To this end, we will use the notation of Parkes, Dash and Jennings [Par01, DPJ03].

the valuation of an agent i for outcome $o \in \mathcal{O}$ given a type Θ_i with the *utility function* $u_i(o, \Theta_i)$. Therefore, agent i prefers outcome o_1 over o_2 if $u_i^1(o_1, \Theta_i) > u_i^2(o_2, \Theta_i)$. This is, the utility function represents the motivations of players in a game and it defines a number or valuation for every possible outcome of the game with the property that a higher number implies that the outcome is more preferred [Par01].

Definition 2 (strategy) *A strategy is a plan of action, scheme, set of moves, program, or method worked out in beforehand that defines the action to be followed by an agent for each possible state of the world in order to accomplish the agent's objective in a given game.*

A strategy must be *complete*, meaning that it should define an action for every contingency, including those that may not be attainable in equilibrium. Now, we are getting closer to the formalisation of a mechanism. The *strategy space* Σ_i denotes the set of all possible strategies available to an agent. Thus, let $s_i(\Theta_i) \in \Sigma_i$ designate the strategy selected by agent i given the type Θ_i. The *outcome rule* $g(\sigma) \in \mathcal{O}$ for $\sigma = (\sigma_1, \ldots, \sigma_{|N|} \in \Sigma^N$ defines the outcome that the strategy σ has for an agent.

Definition 3 (mechanism) *A mechanism Γ is a tuple $< \Sigma_i, g(\cdot) >$, given the strategy space Σ_i and the outcome rule $g(\cdot)$ so that $g(\sigma)$ defines an outcome for each possible strategy σ, where $\sigma \in \Sigma_i$.*

In other words, a mechanism determines the outcome for each possible strategy. If we add the types of the agents to the mechanism then we have a *game*.

Definition 4 (utility in a game) *Let $u_i(s_1, \ldots, s_I, \Theta_i)$ denote the utility of agent i at the outcome of the game, given preferences Θ_i and strategies $s = (s_1, \ldots, s_I)$ selected by each agent [Par01].*

This is, the utility u_i in a game is computed taking into account first the preferences over different outcomes in the world (i.e. the type Θ_i), and second, both an agent's own strategy and the strategies of other agents. Therefore, the classical game-theoretic agent will select a strategy that maximises its utility (which is computed on its type, beliefs about other agent's behaviours and structure of the game). If every agent follows this behaviour and also selects a strategy that maximises its utility, then the game will present a *Nash equilibrium* [Nas50].

Definition 5 (Nash equilibrium) *A set of strategies $s = (s_1, \ldots, s_I)$ selected by a set of agents $(a_1, \ldots, a_I)$ is in Nash equilibrium if every agent a_i maximises its expected utility with strategy s_i given its preferences and the strategy of every other agents.*

Hence, a set of strategies, one for each player, is in Nash equilibrium if no player has an incentive to unilaterally change their action. This is, players are in equilibrium if a change in strategies by any one of them would lead that player to earn less than if it remained with its current strategy.

Nash equilibrium is only one of the solution concepts provided by Game Theory in order to compute the outcome of a game (with the assumptions done so far: rational self-interest agents making suppositions about others' preferences). A stronger solution concept than the Nash equilibrium is a *dominant strategy* equilibrium. Here, every agent keeps the same utility-maximising strategy for all possible strategies of the rest. Therefore, a strategy is dominant if it is always better than any other strategy, regardless of what opponents may do. If a player has a dominant strategy, then it will always play it in equilibrium. Also, if one strategy is dominant, than all others are dominated.

Definition 6 (dominant strategy) *A set of strategies $s = (s_1, \ldots, s_I)$ selected by a set of agents $(a_1, \ldots, a_I)$ is in dominant strategy equilibrium if every agent a_i maximises its expected utility for all possible strategies of other agents. Thus*[18]*,*

$$u_i(s_i, s_{-i}, \theta_i) \geq u_i(s_i^{'}, s_{-i}, \theta_i) \qquad \text{for all } s_i^{'} \neq s_i, s_{-i} \in \Sigma_i$$

The next solution concept is weaker. This time, an *ex post Nash* strategy maximises the agent's expected utility regardless of other agents (thus, so far exactly as in a dominant strategy) as long as all other agents also play an equilibrium strategy.

Definition 7 (Ex post Nash) *A set of strategies $s = (s_1, \ldots, s_I)$ selected by a set of agents $(a_1, \ldots, a_I)$ is in ex post Nash equilibrium if every agent a_i maximises its expected utility when all agents play an equilibrium strategy. Thus,*

$$u_i(s_i, s_{-i}, \theta_i) \geq u_i(s_i^{'}, s_{-i}, \theta_i) \qquad \text{for all } \theta_i \in \Theta_i$$

Finally, a fourth solution concept is the *Bayesian Nash equilibrium*. Here, an agent's utility-maximising strategy selection relies strongly on its beliefs on others' valuations.

[18] We will use an additional notation $s = (s_1, \ldots, s_I)$ for the set of strategies of all agents and s_{-i} for the same set excepting the strategy of agent i.

Definition 8 (Bayesian-Nash) *A set of strategies $s = (s_1, \ldots, s_I)$ selected by a set of agents $(a_1, \ldots, a_I)$ is in Bayesian Nash equilibrium if every agent a_i selects a utility-maximising strategy in equilibrium to the utility-maximising strategies that she expects the rest to select. Thus,*

$$u_i(s_i(\theta_i), s_{-i}(\cdot), \theta_i) \geq u_i(s_i^{'}(\theta_i), s_{-i}(\cdot), \theta_i) \quad \begin{array}{c}\textit{for all } s_i^{'}(\cdot) \neq s_i(\cdot) \\ \textit{and all } \theta_i \in \Theta_i\end{array}$$

Dash, Parkes and Jennings illustrate some of the aforementioned solution concepts in the following example [DPJ03]:

> An example of a *dominant-strategy* implementation is the second-price (Vickrey) auction, where the auction clears for the second-highest price. In this case, the dominant strategy is for an agent to truthfully reveal its valuation for the item.
>
> An example of an ex post Nash implementation is the English auction, an ascending-price auction in which the ask price is ϵ above the current winning bid [MM87]. A straightforward bidding strategy is to bid at the ask price p whenever $p \leq v_i$ for value v_i. This is an *ex post Nash equilibrium*. That is, as long as other agents are also straightforward, an agent can do no better, whatever the other agents' values. However, straightforward bidding is not a dominant strategy equilibrium. Consider another agent that conditions a "crazy" strategy such as "I will bid to $1 million" if the price hits a particular target value. In this case, an agent should submit a jump bid past this target price to prevent this strategy from triggering. Preventing jump bids in the English auction makes straightforward bidding a dominant strategy.
>
> An example of a *Bayesian-Nash implementation* is the first-price sealed-bid auction. Given a symmetric equilibrium of agent types with values that are identically and independently distributed, $v_i \sim U(0, 1)$, the symmetric Bayesian-Nash Equilibrium is for agents to play $s_i^*(v_i) = (|N| - 1)v_i/|N|$.
>
> All of these auctions implement the efficient allocation and are revenue-equivalent in equilibrium.

Now, we can already add some information about the auctions depicted in Table table 2.2, namely the strategies that could be interesting for an agent playing in each one, as shown in Table 2.3.

Auction Format	**Strategies**
English Auction	In the quintessential auction, the dominant strategies for agents is to bid a little bit above the current highest bid and continue doing so if overbid, until the valuation is reached. Some Internet Auctions already offer an "automatic bidding" option to enable this possibility.
Dutch Auction	There is no dominant bidding strategy in this auction. Since the bidder has no direct information on others' valuations, it must rely on prior beliefs about the other's valuation. The analysis of strategies in Dutch auctions can be found in [Mil89].
FPSB Auction	This auction is strategically equivalent to the Dutch Auction because the bidder does not have information on other's valuation either. The agent must act using only its own valuations and beliefs on what the others may bid. One possibility is to bid less than the user's valuation. Unfortunately, there is no rule of thumb that specifies how much less; it just depends on the bidder.
Vickrey Auction	The dominant strategy in Vickrey auctions is to bid the user's true valuation and, thus, this implementation allows us to make efficient decisions. This aspect is true for Vickrey auctions where the value of the auctioned good is private, meaning that the participant doesn't know the valuation of the others (in opposition to public value, where the value is public knowledge).

Table 2.3: **Strategies in some popular auction formats. –** Based in [HJL03]

This equilibrium concepts focus the game from the point of view of the participants. Yet we have named before two kind of conflicting interests: among the bidders and among the bidders and the auctioneers. Nash equilibria, dominant strategy, and so on, refer to the first one, whereas *social choice functions* (SCF) refer to the second.

Definition 9 (social choice function) *Let social choice function* $f : \Theta_1 \times \ldots \Theta_I \rightarrow \mathcal{O}$ *define a desired outcome for each possible set of agent types. Mechanism* $\Gamma = (\Sigma_1, \ldots, \Sigma_I, g(\cdot))$ *implements social function* $f(\theta)$ *if* $g(s_1^*(\theta_1), ...s_I^*(\theta_I)) = f(\theta)$*, for all* $(\theta_1, \ldots, \theta_I) \in \Theta_1 \times \ldots \Theta_I$*, where strategy profile* $(s_1^*, ...s_I^*)$ *is an equilibrium solution to the game induced by* Γ *[Par01]*[19]

Hence, with the social choice function f the designer describes the preferred outcome from the system [DPJ03]. Therefore, when the designer has outlined an SCF, the problem is what incentives can be given so that agents select the strategies that implement that SCF.

In this context, a *direct revelation* mechanism (DRM) is one in which the only available actions to agents are reporting their types. In words, their strategy space is restricted to making direct claims about their valuations. Furthermore, an *incentive compatible* mechanism (ICM) is a direct-revelation mechanism if the best strategy for an agent is to tell the truth. For instance, the Vickrey auction is an incentive-compatible[20] direct-revelation mechanism (in its single-item format) since the best strategy available for each agent is to bid truthfully their valuation for the auctioned good. With these last three concepts in mind, we can go on to define what Parkes calles "an important tool for the theoretical analysis of what is possible, and of what is impossible, in mechanism design" [Par01].

The *Revelation Principle* states that any mechanism can be designed as an equivalent truth-telling, direct revelation mechanism to achieve a Nash equilibrium outcome. Or, in other words, that any mechanism Γ has a direct-revelation, incentive compatible mechanism with the same outcome. The revelation principle was first formulated by Gibb [Gib73] for dominant-strategy equilibria, and extended afterwards by Green and Laffont [GL77] and Myerson [Mye79, Mye81]. Formally[21] it can be defined as:

[19] As in [Par01], we skip defining the equilibrium concept. It may be any of the ones presented previously, generally as strong a solution concept as possible.

[20] Actually it is strategy proof, since truth revelation is a dominant-strategy equilibrium [Par01].

[21] For a detailed proof of the revelation principle theorem see for instance [MCWG95] or [Par01].

Definition 10 (revelation principle for dominant strategies) *For any mechanism, $\Gamma =< \Sigma_i, g(\cdot) >$ that implements the social-choice function $f(\cdot)$ in dominant strategies, $f(\cdot)$ is truthfully implementable (incentive compatible) in dominant strategies.*

According to Dash et al. [DPJ03], the revelation principle is important in Mechanism Design for two reasons:

- *Theoretical*: It allows a focus on incentive-compatible direct revelation mechanisms for the development of impossibility and possibility results.
- *Practical*: The properties that incentive-compatible direct revelation mechanisms can satisfy can provide a normative guide for the outcome and payments that a realised implementation must compute. This mechanism need not itself be a direct revelation mechanism and can have better computational properties than the original mechanism.

One could consider that the revelation principle renders non-incentive compatible mechanisms uninteresting. Nevertheless, in many cases incentive compatible mechanisms are simply not feasible from a computational point of view [DPJ03, CS03]. That is, computational mechanism design is not equal to mechanism design since the first exists in a dimension that requires considering new aspects (such as computability, performance, feasibility, communication, and so on [LS01, KDMT01, BN02, BNS03]).

Both computational mechanism design and mechanism design are nowadays subject of a steadily increasing research. Not only as a powerful tool for designing efficient multi-agent decision systems taking advantage of theoretical constructs of Game Theory, but also as reinforcement of e-commerce. In this way, the possibility of organising auctions online has boosted interest in auctions and specially on updating game theory's focus [RZ94, San93a, Par01, DPJ03], because it draws a new, exciting framework in which designing optimal and efficient auctions poses many unexpected challenges. For instance, He and Jennings [HJL03] describe three possible scenarios enabled by the Internet and agent-mediated e-commerce:

- *Scenario 1 - Finding closest match to buyer's requirements*: A buyer decides that they would like a holiday in one of the Greek islands, they would like to go next Friday, they would like to fly from London, and that the total cost should be less than 300 pounds.

Their software agent is instructed to go and find out what is available and to report the options back to the user who will make the ultimate choice. In order to fulfill this objective, the buyer agent determines those e-markets that deal with leisure activities. From those, it tries to find out holidays that meet the specified requirements. However, it finds no appropriate fixed price offerings and after observing the outcome of several online auctions, it decides that it will be very unlikely that it will be able to meet all of these requirements. It therefore decides to relax some of the user's constraints and tries to find holidays that are similar. The agent decides to relax the user's stated requirements in the following way: it looks for holidays to the Greek islands that leave anyday next week, that leave from non-London airports in the United Kingdom next Friday, and that cost up to 400 pounds. With these new requirements in place, the buyer agent returns to the relevant e-markets, collects the offerings that satisfy these new requirements, and returns them to its user with an explanation of why it acted in this way.

- *Scenario 2 - Acting across multiple e-markets*: A buyer decides that they would like to purchase a new laptop computer; they want a reasonably high specification, are prepared to pay for a good quality brand name, but it must be delivered within a week. Their software agent is instructed that they are prepared for the agent to find the most appropriate model, negotiate the best potential deal available, but that the user would like to make the final choice about purchase. In order to fulfill this objective, the buyer agent determines those e-markets that deal with selling computer equipment. From these, it selects those e-markets that offer products that meet the user's specification. In order to determine those machines that fit the specification, the buyer agent examines the sites of a number of computer manufacturers to determine the latest specification information and to determine an approximate price to pay. Armed with this information, the agent formulates a strategy for making a deal. The agent knows the maximum price it needs to pay (this will be the minimum of the cheapest fixed price offerings that are available in the catalogs). From this baseline, the agent tries to negotiate directly with several of the suppliers to see if they are willing to reduce the price (or bring forward the delivery time). In parallel to this, the agent tracks a number of online auctions to see if the same good can be purchased more cheaply (it will not actually bid in the auctions since submitting a bid would constitute a commitment on behalf of its buyer). When it has completed its negotiations (or before if a very good deal appears in an auction), the buyer agent reports back a ranked list of purchasing options to its owner. The owner then makes their choice and instructs their agent to complete the deal (including arranging payment and setting the delivery time and place).

- *Scenario 3 - Coalition formation*: A bakery agent receives a request for tender from a supermarket agent who wishes to purchase 500 iced buns a day throughout the summer period. The bakery agent has sufficient capacity to make 300 buns per day. However, the bakery would like to set up links with the supermarket and so is keen to see if it can fulfill the order. Thus, rather than simply turning the order away, the bakery instructs its agent to search for a partner who will produce the remaining 200 buns for the rest of the summer period. In order to achieve this, the bakery agent contacts all the other sellers present in e-markets that offer iced buns. The bakery agent indicates it has a demand for 200 buns per day for the summer period and asks whether any of the other bakeries would like to join in a partnership with it to meet the supermarket's need. A number of potential collaborators come forward. The bakery agent then conducts a series of negotiations with these agents in order to set up the terms and conditions of the partnership. Eventually, a deal is reached and the bakery agent reports details of the arrangements back to the bakery.

Focusing on the auctions domain, agents have shown to be more effective than human bidders. For instance, Gode and Sunder show in [GS93] how agents outperform human counterparts in a particular auction setting (the CDA). Das et al. demonstrate in [DHKT01] that agents with "Zero Intelligence" can partially replace humans' experience and learning in auctions. Specifically, they conclude that the high market efficiency typically observed in continuous double auction experiments with human subjects is due to the structure of the auction and not to learning. Moreover, agents present a further advantage that human cannot meet, *ubiquity*. Agents can compete simultaneously in multiple auctions, be in many places at the same time. This possibility implies that the agent is able to compare auctions, better search for a certain good, asses a market's valuation, and help the transaction price be as close as possible to the equilibrium price[22] [PBB01].

Now, so far, we have been dealing with a concrete kind of auctions. Specifically, with auctions where generally only commodities of one type are traded. If we allow the auctioning of several goods of different types simultaneously, buyers may express their preferences not only on the price they want to pay, but also on they interest of buying two or more of the items at the same time. This is, a combination of them. Literature on the subject has coined the term *combinatorial auction* for this new class. We now go onto to discuss this.

2.2.2 Combinatorial Auctions

In the previous section, we have been discussing auction formats in which items are sold *individually*. That is, regardless of the units sold, items are not linked to each other in any way. Consequently, there are no expressions of interest in buying, for instance, two items *at the same time* and the valuation for this is different than the sum of buying the constituent items separately [PJ05]. Such a setting may appear in, for instance, electricity markets (as the one issued in this Dissertation), "equities trading, bandwidth auctions [MM96, McM94], markets for trucking services [San91, San93b, San00b], pollution right auctions, auctions for airport landing slots [RSB82], and auctions for carrier-of-last-resort responsibilities for universal services [KS00]", as detailed in [San02].

Currently, there exist a number of auction mechanism that could deal with the features described above. Sandholm, whose work has been seminal in this area, gives a detailed account of them in [San02]:

- *Sequential Auction Mechanisms*. In this auction setting, items are auctioned one at a time. Therefore, the method for determining the winner is easy: it simply consists of selecting the highest bidder for each item separately. The problem arises if a bidder wants to buy more than one item together (e.g. she only wants item A iff she also gets item B, and would even pay more than her valuation of item A plus her valuation of item B for them, if this assures that she gets both). This implies speculation on others' behaviour in future rounds (items) and may cause the bidder buying only a part of the bundle (when, as explained, the original idea was buying *the whole* bundle or nothing).
- *Parallel Auction Mechanisms*. This auction setting provides an alternative to sequential auctions. Items are auctioned simultaneously and bidders may bid

[22] The market price at which the supply of an item equals the quantity demanded.

for many items openly. The only advantage of this mechanisms is that the bidder may get an idea of other's valuation and, therefore, bid in consequence. Nevertheless, the problem of sequential auctions remains.

- *Methods for fixing inefficient allocations*. These strategies try to overcome the problems of the aforementioned auction mechanisms. One approach consists in setting up a second market where bidders can re-auction items they may have bought and do not want (e.g. if a bidder wanted item A together with B and only gets B, A will try to sell B). Unfortunately, this amendment does not assure an economically efficient allocation in general or, if it does, it takes a prohibitively high costs. Another approach has been put into practice by the Federal Communications Commission, where they allow bidders to retract their bids in case they don't get the combination they want. The items are then offered in a new auction and if the new price is lower than the old one, the retracted bidder must pay the difference.

The fourth item in this list is combinatorial auctions [dVV03], which both removes the shortcomings of looking ahead and the inefficiencies stemming from uncertainties [San02]. Their major distinctive aspect has been already cited: "buying a bundle of items is valued differently than the sum of buying the constituent items separately" [PJ05]. Think, for instance, about the auction of some same-sized plots of land, depicted in Fig 2.14.

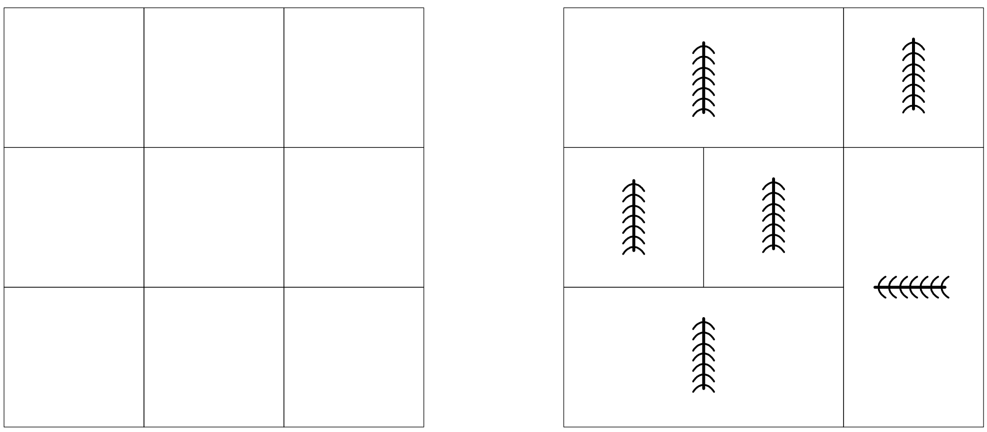


Figure 2.14: **Plots of land to auction and possible result if bundles are permitted.** – Bidders will have a higher valuation of wining a bundle of neighbouring plots since they may share the irrigation system so they don't need to build it specially for each plot.

The figure on the left shows all the plots before the auction and the figure on the right shows a possible outcome of the auction. Considering that having two neighbouring plots may spare them from building two separated irrigation systems, the valuation for buying them *together* in a bundle is bigger than buying them separately and, therefore, a bidder is likely ready to offer a higher amount for buying two neighbouring plots together. In our problem domain, the reason for buying two items together is that the supplier offers a discount if this happens.

Although combinatorial auctions may potentially increase the auctioneer's benefit [FLBS99, SSGL02], they lack the efficient clearing algorithms for settling the price to be paid, determining the winner and fixing the quantities of allocated goods. This is generally known as the *winner determination problem* (WDP) [SSGL01b, PU00] (also as *the bid evaluation problem* [EGKL01]). If the combinations included in the bids are the same for all the bids, then the winner determination problem can be solved trivially, as in the non-combinatorial auctions, by choosing the highest bid[23]. The problem arises when the bidders place bids as depicted in Table 2.4.

Items:	Item ♢	Item ♡	Item ♣	Item ♠	Bid
Bidder 1:	♢		♣		for 3 €
Bidder 2:			♣	♠	for 3 €
...					
Bidder n:	♢	♡		♠	for 6 €

Table 2.4: **Example of bidders and their bids in a Combinatorial Auction** – Finding the optimal allocation that maximises the seller's pay-off is no more non-trivial.

Unfortunately, the combinations expressed in the bids generally overlap. Therefore, finding the optimal allocation from items to bidders so that the auctioneer maximises her revenue is conceptually equivalent to the *weighted set-packing problem* [Kar72] and, thus, *NP-complete* (or *NP-hard*, i.e. there is no algorithm that solves this problem in polynomial time) [RPH95].

Now, we present here the formalisation of the WDP using Sandholm's notation in [San06b, SSGL01a]:

Let $N = \{1, \ldots, n\}$ be the set of bidders participating in the auction. Let $M = \{1, \ldots, m\}$ be the set of items to be auctioned. Let S be a bundle of items so that $S \subseteq$

[23]We assume from now on that the auctioneers clears the auction looking for the allocation that maximises her revenue.

M. Let $v_i(S)$ denote the bid that bidder i places for bundle S. Let $x_i(S) \in \{0, 1\}$ denote the allocation of items so that x_i is equal to one iff bidder i gets bundle S. An allocation $(x_i(s)|i \in N, S \subseteq M)$ is said to be *feasible* if it allocates every item just once:

$$\sum_{i \in N} \sum_{S \subseteq M, S \ni j} x_i(S) \leq 1 \qquad \text{for all } j \in M$$

and at most one subset to each bidder:

$$\sum_{S \subseteq M} x_i(S) \leq 1 \qquad \text{for all } i \in N$$

Thus, the winner determination problem can be defined as, given the bids $v_i, i = 1, \ldots, n$, the problem of computing:

$$x \in argmax(\sum_{i \in N} v_i(S) x_i(S) | \textit{ x is a feasible allocation})$$

One may think that the reverse auction is equivalent to the forward auction in terms of the winner determination problem. However, as Sandholm et al. show in [SSGL02] the results analysing the complexity of determining the winner in each case are quite different. For instance, in the case of *free disposal* (i.e. that non-allocated resources can be disposed without a cost). This property can be formalised as [Cra06]:

$$\text{if } v(S \bigcup T) \geq v(S) \qquad \forall S, T.$$

Whereas in the forward auction, both in the single and in the multi-unit case[24], the solution cannot be approximated, it is possible for reverse auctions, even in its multi-unit case. Unfortunately, in case of no free disposal, even trying to approximate the single-unit reverse combinatorial auction winning allocation is a hard problem [SSGL02].

Lately, many researches have tried to overcome the complications posed by the winner determination problem. These attempts can be classified into the following categories [San02]:

[24] In a single-unit format, one item, good of service is auctioned. In the multi-unit case, the auctioneers sells more than one unit of the same item or good.

- *Enumeration of exhaustive partitions of items*: A *partition* is a bunch of subsets of items so that each item is included in at most one of the subsets. The advantage of using partitions is that, given a partition, it is trivial to determine the winning bids. In this way, an *exhaustive partition* presents all items included in only one subset of the partition. So, Figure 2.15 illustrates the enumeration of exhaustive partitions for an example with four items. As can be seen, the main problem arising from this approach is that the number of exhaustive partitions grows exponentially and it becomes intractable very soon (according to Sandholm, above a dozen items [San02]).

- *Dynamic Programming*: This technique allows a more efficient search in the exhaustive partition space [RPH98], such as the one depicted in Figure 2.15. Sandholm presents a dynamic program for winner determination in [San02]. This program, however, computes even the combinations of bids that have not been submitted and, therefore, is outperformed by other strategies in case where there are a high number of items [San02].

- *Focusing on relevant partitions*: In this approach, the algorithm is focused only on partitions containing combinations that have been bid. The problem can be reformulated as an integer program, but, unfortunately, it does not reduce the complexity of the problem, which is still NP-hard [RPH98, San02].

- *Polynomial-time approximation algorithms*: The objective now is to achieve tractability, though at the expense of reducing the quality of the solution. That is, polynomial-time approximation algorithms try to find a reasonably good relevant partition instead of an optimal one. To this end, some researchers have tailored general polynomial-time approximation algorithms [Hoc97, KY99] to the WDP [San02] and some others have developed new ones [DJ02].

- *Combinations restriction*: Given n items, there are $2^n - 1$ possible combinations. The goal of restricting the possible combinations or the amount of possible combinations tries to make winner determination in these situations tractable (i.e. solvable in polynomial time). Possible strategies include not only restricting the possible combinations per bidder or the amount of items in each combinations, but also allowing only bids in tree structure, introducing families of combinations of items, etc. Nevertheless, bidders may be prevented from bidding for the combination they wanted and, therefore, there is a trade-off between computational speed and economic efficiency [San02].

- *Search Algorithms*: In this approach, the space of the possible combination between items (i.e. exhaustive partitions) is traversed following a

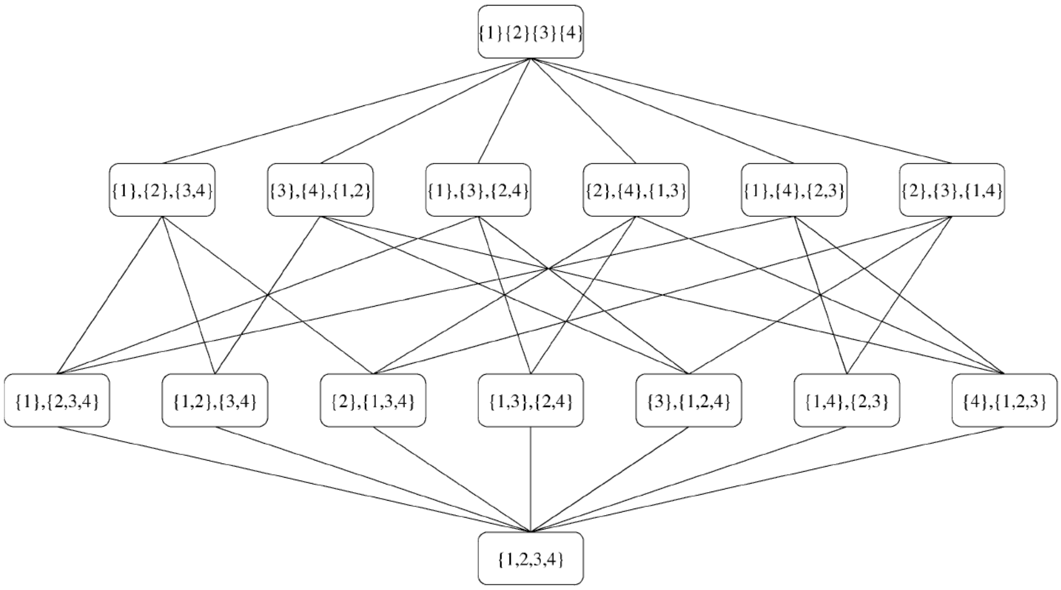


Figure 2.15: **Exhaustive Enumeration of partitions in a 4-item example [San02] –** Space of exhaustive partitions (in a 4-item example). Each vertex represents one exhaustive partition. Followed upward, each arc represents dividing one of the bundles within an exhaustive partition into two. Followed downward, each arc represents merging two bundles within an exhaustive partition into one.

search algorithm until it finds an optimal solution. Search strategies include depth-first search ([SSGL02]), depth-first branch-and-bound search ([FLBS99, SSGL01b], A^* based on [DP85] and [HNR68] and all kind of heuristics to prune the search tree [SS00b, SS03]. All these algorithms have been compared using a variety of randomly generated distributions [FLBS99, LBPS00, ATY00, San02]. A compilation of them can be found in [San06a].

Besides the complexity of the winner determination problem, combinatorial auctions pose some additional problems. For instance, allowing bidders to express their preferences implies having a kind of language that enables this preference elicitation. In normal (non-combinatorial) auctions, this issue has been often ignored but combinatorial auctions demand that it is tackled. Specifically, with $2^n - 1$ possible combinations, the auctioneer or the designer of the auction, must provide bidders with a tool for this purpose. Moreover, by improving bidding languages, winner determination problems will work having better information [BGN03]. We will give here just a brief description on bidding languages. More detailed information can be found in [Nis06, Nis00, BH01].

- *Atomic Bids*: The bid placed by the bidder is a tuple (S, p), where S is a subset of the items and p is the price the bidder offers for the items in S [LOS99]. Atomic bids fail to represent many simple bids, as for instance, simple additive valuation on two items [Nis06].

- *OR bids*: Bidders submit a collection of atomic bids (S_i, p_i), where each S_i is a subset of the items and p_i is the corresponding price the bidder offers for the items in S_i. The goal of the bidder is to win any number of disjoint atomic bids for the sum of their prices. This model cannot represent bids presenting substitutabilities among them [Nis06].

- *XOR bids*: Bidders again submit collection of atomic bids (S_i, p_i), but this time, the goal is to win at most one of them [San03]. With this format, it is possible to represent all valuations [Nis06]. There exist also combinations of OR and XOR bids (namely OF-or-XOR bids, XOR-of-OR bids and OR/XOR formulae bids [Nis00, LLN01]).

- *OR Bids with Dummy Items*: The key idea in this language is using "dummy" items in the bids, without intrinsic value but as a vehicle for expressing constraints. OR bids with dummy items combine the expression power of XOR bids with the simplicity of OR bids, so that algorithms developed using OR bids will be able to use this variant without any change [Nis00, Nis06].

The work on combinatorial auctions presented above has included research on atomic prepositions [GL00, Nis00, DK01, EGKL01, San02], which have to be accepted in their entirety or rejected. There are, however, some cases where atomic bidding cannot offer all the flexibility that bidders require and, thus, the revenue of the auctioneer may not be maximised. For instance, in case of bids that can be accepted partially. Furthermore, XOR bids and the like don't allow the bidder to express explicitly a relation between the price and the quantity. This is not adequate, for instance, when dealing with a good such electricity or water supply, when prices depend on the quantity. To this end, Sandholm and Suri introduced the possibility of bidding with a demand or supply (depending whether the auction is forward or reverse) functions [SS01]. However, as Dang and Jenning point out [DJ02, DJ03], this work is just limited to the multi-unit single-item auction.

Last, but not least, there is still an issue on combinatorial auctions that demands a bit of attention: the mechanism. When designing the combinatorial marketplace, researchers almost automatically go for the Vickrey Auction model (or, in its multi-unit variant, the Vickrey-Clarke-Groves (VCG) mechanism [Cla71, Gro73]) because, as explained in section 2.2.1, it *motivates* agents to tell the truth (i.e. is a incentive-compatible direct-revelation mechanism). Remember that, in non-combinatorial auctions, the intuitive reason for this incentive was that the bidder had no influence on the price she would pay and, therefore, the best strategy was bidding her true valuation. In combinatorial auctions, with the VCG mechanism, the idea is the same. Bidders place their bids on combinations of items. The winner determination algorithm finds the allocation that maximises the revenue of the auctioneer given the agents' values. And then, the winners pay what they have bid minus the so-called *Vickrey discount*. And this amount is exactly the crux of the matter. It is calculated by subtracting the total value computed by the winner determination problem *without* the bidder from the total value *with* that bidder. Hence, they have, as in the non-combinatorial auctions, no influence at all on the price they pay in case of winning. The formalisation, and a more detailed description of the VCG mechanism, as well as further information on possibility and impossibility results of incentive-compatible direct revelation mechanism can be found in [MCWG95] or [Par01]. Critics to the VCG and a discussion on its low (or zero) seller revenue problem can be found in [RTK90, AM06]. For the Ausubel auction as alternative to VCG see [Aus04, Aus06].

2.3 Electricity Markets

In 1990, the UK Government privatised the UK Electricity Supply Industry in England and Wales. In 1996, as stipulated in the European Community (EU) Directive 96/92/EC, the European countries officially decided to develop a single market for electricity [cee03]. This event was the milestone that opened the process of creating the largest competitive electricity market in the world. The integration of energy markets is supposed to lead to greater efficiency and contribute to the security of the supply.

The electric power, or electricity for most consumers, of this supply is generated by utility companies usually using coal, oil, nuclear, or hydropower. It involves the production and delivery of electrical energy in sufficient quantities so business and households can operate according to their demands. Some of the generating capacity is presently based also on renewable energy sources such as solar power and wind power. But their share as part of the total energy system has been rising since the mid seventies and is expected to contribute up to 21% of total energy supply in 2010 [Com05] in the UCTE (Union for the Co-ordination of Transmission of Electricity, the association that includes most of the continental members of the European Union). Nowadays, a 24-hour on-demand, access to electrical power is taken for granted for residents of most developed countries.

Before liberalisation, the structure of the electricity market had territorial monopolies, extensive public ownership, federalised organisational structures, and a lack of a market-pricing mechanism [HA01]. Electric power companies owned the whole infrastructure from generating stations to transmission and distribution infrastructure. The industry was generally heavily regulated, often with price controls, and was typically government-owned. For this reason, electric power was defined as a *natural monopoly* [Haa03].

On the basis of the above mentioned Directive, each EU Member State is obliged to gradually open its national electricity market with the objective of the full liberalisation of the electricity market by the end of 2005 (which has been almost completely achieved [Com05]). Though each Member State operates at its own pace of market opening, trying to harmonise existing rules with measures to accomplish the requirements of the Directive [MJ01]. The EU Directives set out the requirements under which competition can be developed in a fair and transparent way. Opening up electricity production to competition is an important tool to improve the efficiency of the electricity production industry and therefore should benefit all electricity consumers. Competitive forces can drive producers to innovate and operate in more efficient and economic ways. Innovation can lead to lower prices and a better use of energy resources. Cost savings due to increased efficiency gains will lower prices for electricity users. This is the intention that motivates deregulation processes in the European countries but unknown side effects might prevent this outcome [Com04].

Basically, deregulation allows energy consumers to choose their electric energy supplier and therefore dissolves electric utility monopolies. The resulting re-regulation and restructuring of the electricity industry has created opportunities and challenges that need to be addressed to ensure long-term capacity sustainability. The promise and benefit expectations of electricity market liberalisation may need to be tempered by the reality of the process. Market liberalisation of the electricity supply sector depends on many different factors and boundary conditions in the EU.

There are three components of the electricity market, namely generation, transmission to the substations, and distribution system from the substations to the end consumers addressing their electricity demand, as illustrated in Figure 2.18.

Electricity generation is the first of the three processes. As we already mentioned, utility companies typically use coal, oil, gas hydropower, or nuclear power to generate electricity. After generation, the transportation of electricity is split up into two

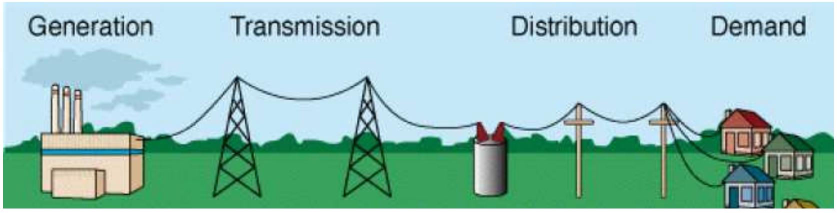


Figure 2.16: **Market components in the electricity market [GAO03]** – The energy generated in the power stations is transmitted by the TSOs and then distributed locally to the customers by the DSOs.

processes, transmission and distribution system in charge of the distribution. The first kind of system consists of transmission lines and substations carrying voltages of between 110 and 400 kilovolts. Substations at points of connection to a distribution system bring the voltage down to distribution levels. The transmission system operator (TSO) is the entity responsible for operating the high-voltage transmission grid, to which electricity producers deliver their production.

Each TSO typically has several interconnections with the transmission grids of neighbouring utilities (the so-called *international tie lines*), because cross-border imports and exports of electricity flow via the transmission grid [Com04]. An interconnection provides a link (lines, cables and equipment, including transformers, etc) that may be used to convey electrical energy in either direction between networks, power stations, or between power stations and networks [Vas03]. In the EU there exist four main TSOs included in the European Transmission System Operators (ETSO): UCTE, NORDEL (Nordic Countries), the UKTSOA (United Kingdom), and the ATSOI (Ireland) as shown in Figure 2.17.

After transmission, the distribution companies tap from the grid via substations and transformers that lower the voltage level to distribution levels. The distribution system operators (DSO) are the entities responsible for operating the medium and low voltage distribution lines. Power lines are used for the lower voltage electricity flow from transmission facilities to commercial and residential customers. In most countries, there is one transmission system operator and several distribution system operators [Com05].

Further, the electricity system in most European Countries has traditionally been vertical-integrated. That is, one large utility owns and operates all three primary aspects of electricity operation, generation, transmission, and distribution, in a given area of service. All the functions performed by local utilities to produce high-quality, reliable electric service (power production, transmission, distribution, voltage regu-

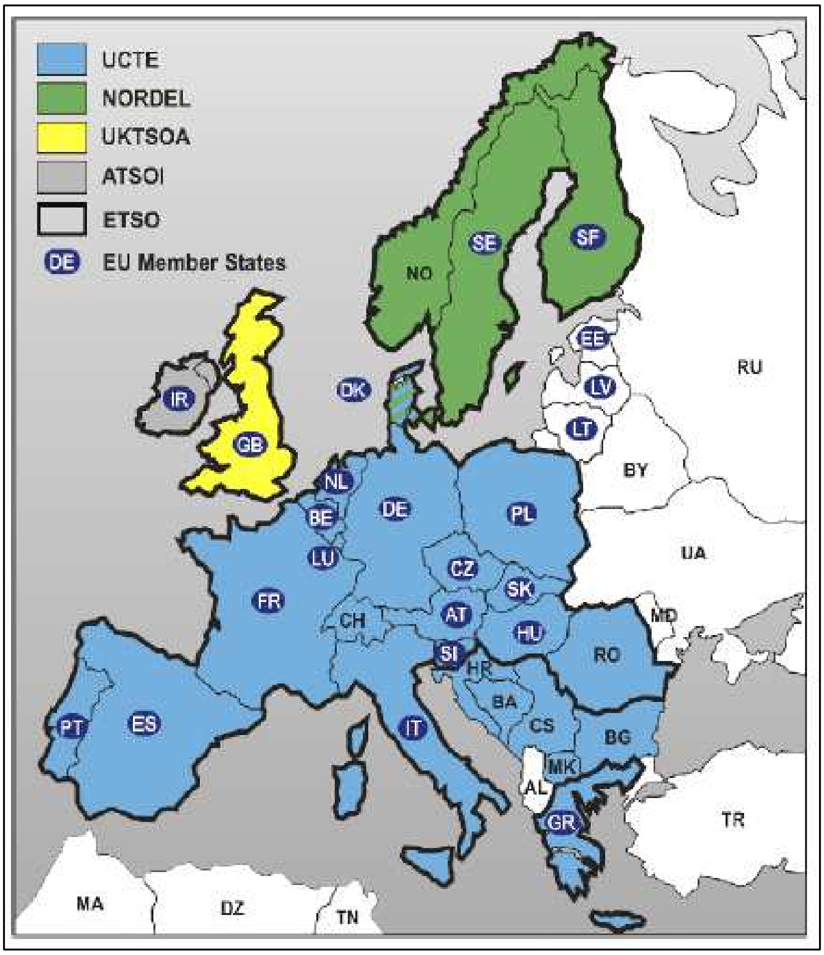


Figure 2.17: **Transmission System Operators included in the European Transmission System Operators (ETSO) –** The ETSO is composed of the UCTE, NORDEL, UKTSOA, and the ATSOI.

lation, etc.) were sold as a package and only bundled services were available. Thus, solely local-utility-produced energy could be delivered over its transportation system. In conclusion, under the former monopoly model these vertically-integrated companies had exclusive rights to supply electricity to residential, commercial and industrial retail consumers within a defined geographic area.

The deregulation splits or *unbundles* the electricity package, separating the three elements into different products that can be marketed and traded independently from each other. Transmission must be offered on equal terms to all market participants so that customers are able to purchase energy from suppliers other than the local utility company [Jos03].

Still, there is a risk that transmission owners may discriminate in favour of their own group companies when granting access to the net-work. To prevent this situation the EU Directive requires Member States to take three basic preventive measures:

- ensuring management unbundling of the transmission system operator
- ensuring accounting separation of transmission and distribution activities from other parts of the company
- ensuring that appropriate mechanisms are in place to prevent confidential information from being passed by the transmission system operator to other parts of the company.

In order to ensure fair access for all market players in the network, it is necessary that, during the management of unbundling, confidential information does not pass from the transmission system operator to other parts of the group. It is an essential precondition to allow competition in generation and distribution to obtain more efficient and effective operations. Unbundling of accounts also will increase transparency in the operation of electricity undertakings. An alternative to the management unbundling approach of the Directive is to legally separate the transmission system operator from the vertically-integrated company. It will then become a separate operation and function independently from other electricity companies. This approach is the most effective in ensuring that discrimination does not take place, and it's been followed by most European countries, like in Austria where the *Verbundgesellschaft* [25] is established as a separate entity [Com05].

The complementary relationships among generation, transmission, distribution and system operations must not be overlooked when pursuing deregulation. This

[25] http://www.verbund.at/at/

means that there might be negative outcomes occurring because of the unbundling processes. Furthermore, the unbundling of a network relationship may bring about a loss of social welfare if the positive effects of competition are less than the negative effects of the vertical unbundling. This effect, established by Antoine Cournot in 1838, is known as the *Cournot Principle* [Cou38].

In most cases, the restructuring process caused by Deregulation involves then separating the electricity generation and retail from the natural monopoly functions of transmission and distribution, and thus, the establishment of a *wholesale electricity market* for electricity generation and a *retail electricity market* for electricity retailing. As already noted in the first chapter, we focus here on retail markets.

Electricity retailing is the final process in the delivery of electricity from generation to the consumer whereas the wholesale market exists when competing generators offer their electricity output to retailers. Electricity retailers have to be able to perform billing, meter reading, and customer management via, for example, a call centre that can handle energy distribution through the use of system contracts and reconciliation agreements. Trading on power exchange markets and hedging contracts in risk management can also be facilitated.

Figure 4 shows an overview of producers, suppliers and consumers participating in wholesale and retail markets.

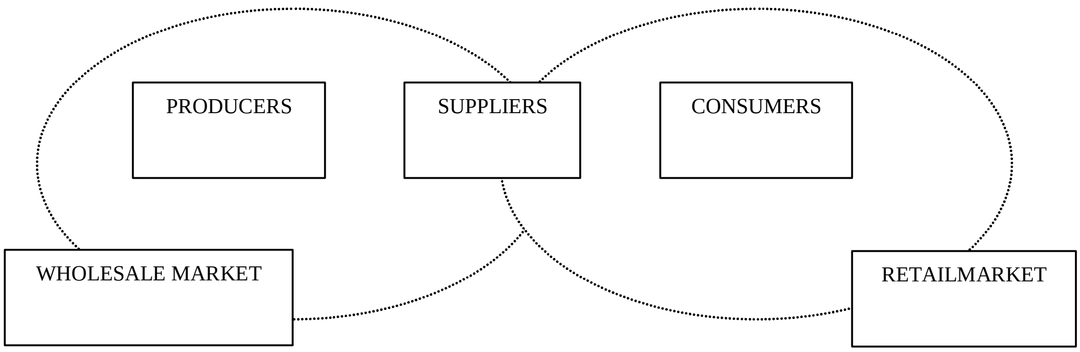


Figure 2.18: **Overview of the electricity market in general –** In the retail market, electricity is delivered to consumers by suppliers and in the wholesale market producers and suppliers interact with each other.

Now, by establishing a retail market, Deregulation aims at setting a market-based price for electricity. This process works in the following way: During most time periods in the electricity spot market, the generation price of electricity will be set by the operating costs of the most expensive generating unit needed to meet demand or what in economics is referred to as the “marginal cost” of production. In general, a supplier will not be willing to sell power below the market price of the most expensive facility operating at a given time, because consumers will be willing to pay a higher price [Haa03].

Similarly, consumers will be unwilling to pay more than the cost of the most expensive operating available generator, since other suppliers will be offering lower prices. With prices set to marginal costs, the market will clear which means that all suppliers willing to provide power and all consumers willing to purchase power at the market price will do so [SN01].

During periods of extremely high demand (peak demand), typically on very hot summer days (or cold winter days), when the demand for electricity approaches the available generating capacity, prices rise above the operating costs of the most expensive generator operating (including fuel costs). In the long-term view, the target is to allow electricity prices to reflect the long-term marginal cost of the society it operates in at all times [Haa03].

From the point of view of production, we can divide this demand into three *loads* [Uni81, Pal01]: *base load*, *intermediate load*, and *peak load*. The base load, devoted to supply the basic energy demand of a region, is constant during all the time and its energy is typically obtained from coal, hydroelectric or atomic power plants, whose production is similarly non-stop and steady. The intermediate load covers seasonal and daily deviations and is provided by the so-called *daylight power plants* or *5-days power plants* because of their usage [Pal01]. Finally, there are still occasional sudden load increments (which may be regular as the so-called *noon peak* or totally random), usually supplied by gas turbines, as shown in Figure 2.19. The reason for this is that conventional power stations like caloric or hydroelectric power stations react very slowly. Only gas turbines can react quickly enough to cover sudden demand peaks [IZE94]. An example for the proportions of these loads can be found in [Uni81] (base load 70%, intermediate load 25% and peak load 5%).

Nevertheless, the liberalised electricity market presents gigantic dimensions. Only in the the 21 countries of the continental European Union (this is, excluding Nordic countries, United Kingdom and Ireland), 450 million customers get daily their electrical energy in a common transmission grid. Moreover, according to the report issued in 2002 by the SYSINT forum [SYS02], over 2100 TWh were delivered in the 21 countries of the UCTE in the year 2001. Furthermore, within this group of countries the load rose approximately 3,96% from year 2000 to 2001, whereas the production capacity increased only 2.75% [UCT02a]. This is, the overall energy consumption increases at a faster pace than production does [Fas95, Boe96]. This

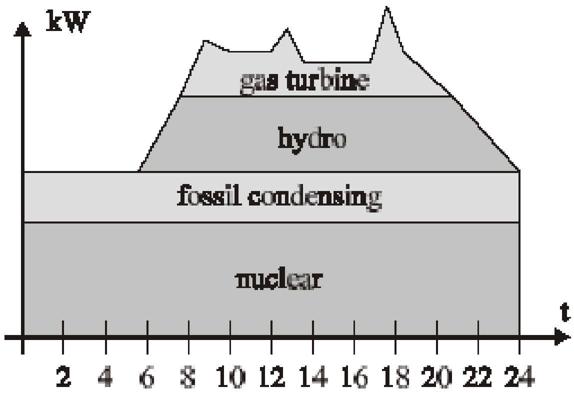


Figure 2.19: **Load composition [Pal01] –** The demand load is supplied with a combination of energy sources, as depicted in the Figure. The characteristics of each source determines the kind of demand they supply to.

difference projects reasonable doubts on the reliability of the energy supply in the near future.

The consequences of not being able to support the demand of energy are seriously worrying: brownouts or blackouts[26] and the subsequent economical loses.

The following example illustrates the dangers that stem from deregulation. In this way, the *California crisis*, in the summer of 2001, emphasised the painful consequences of a chronic supply/demand imbalance and market manipulation [Cro02]. Electricity prices were high in California partly because of the regulated market, by assuring producers of a high rate of return on their investments, provided incentives to build too much generating capacity. When California opened its electricity generation market to competition, policy makers hoped that competition would reduce electricity prices but they also imposed a price ceiling to maintain stable retail prices [HA01].

During 2000, the rising energy prices and the reduced availability of capacity decreased electricity supply in California. Rising gas prices increased the cost of production for the plants that mainly rely on natural gas as fuel. Additional strong economic growth, in particular because of growing computer-based businesses in the

[26] The difference between brownouts and blackouts is that in the latter, no power at all is available and in the former, some power supply is still retained (though, the minimum level specified for the system). Moreover, TSOs may plan brownouts, called then *voltage reductions*, in order to prevent a blackout.

Silicon Valley area, and severe weather conditions boosted electricity demand. The market-clearing price was higher than the price ceiling and could not be charged to the consumers. Furthermore, there were problems on the supply side because polluting plants were idled and old power plants (55% of California's plants were more than 30 years old) operated less efficiently [HA01].

With the price ceiling from the government in place, consumers tried to purchase much more electricity than producers were willing to sell at the ceiling price. Critics warned that blackouts might result because demand and supply could not match at the fixed level of the market price. But California's utilities were legally obliged to supply all the electricity consumers wanted to purchase at the ceiling price. To do so, utilities were forced to pay a much higher price for electricity on the open market. Because the utilities did not quite succeed in obtaining all the electricity that customers want at the ceiling price, the result was a combination of short-ages and utilities paying higher prices for electricity than they could sell it for to their own customers. In the tight supply situation, some generators were shut down because of unscheduled power-plant maintenance [HA01].

By the end of 2000, California utilities were paying a wholesale spot price of about 40 cents per kilowatt hour while they were only allowed to sell it for about 10 cents per kilowatt hour to their customers. In summer 2001 the disruption to businesses and homes as a result of the ongoing blackouts and extreme prices was enormous. California's failure to allow retail prices to rise to reflect market conditions has put a financial burden on the utilities. In addition, low prices discouraged the development of additional supply while encouraging customers to continue using electricity [HA01].

Now, the problem is not (only) that the demand increases rapidly, but that some of the peaks cannot be covered by the existing supply capability. To prevent this worst-case scenario, there are mainly three solutions: finding new energy sources, distributing the demand to smooth consumption peaks or somehow managing to keep electricity on stock. On the one hand, renewable energy sources are promoted and supported by the European Union and the respective national governments but they still cannot compensate to the difference between load and capacity increment (they constituted approx. the 4% of the UCTE generating capacity in 2003 [UCT02b]). Building new power plants is also an unacceptable solution for the society and bad from an ecological point of view (increasing of the CO_2 emissions).

On the other hand, producers may try to store energy. For instance, when the energy consumption is low, generators may continue producing to load this surplus electricity into accumulators or batteries. Batteries are, however, almost not in use anymore [Pal01]. In this way, a more extended method to store energy is to pump water into reservoirs located on top of a mountain for high head power stations [Mos91] [Pal01]. Sometimes ordinary hydroelectric power plants may be also employed for this use as detailed in [Eme95]. Nevertheless, all these techniques are very costly so, at the end, stored energy becomes much more expensive than on-demand generated energy [Pal01].

Therefore, there is only one sensible alternative left, namely trying to optimise the energy demand (if not possible to reduce it). This alternative is not only sensible, but economic as well. For the utilities, influencing demand is cheaper than building new generators to match this increasing demand [Wir94].

In this way, Demand-Side Management (DSM), as an aspect of integrated resource planning (IRP, [Haa95]), is a term used for all kinds of energy management carried out at consumer's side [Pal01] (as opposed to Supply-Side Management, SSM, which is not the focus of this work).

The basic forms of DSM include, for instance, installing sensors that detect when nobody is in the room and eventually turning off the light. Such measures have been already successfully implemented by many companies around the world. In order to remain within the context of this work, we will concentrate here on DSM techniques using AI techniques.

There has been a surge of research in this area lately, most of them are concentrated on properties of the underlying multi-agent system and slightly out of focus to our needs. Moreover, only the work of Palensky et al. [PDPR97, Pal01, PRD03, SPL+05] is explicitly directed to remove the shortcomings posed by Deregulation (or seen in a positive way, to exploit the new opportunities it brings). Specially interesting is his PhD thesis, where he addresses a genetic algorithm as an optimal method to find the proper demand schedule. Nevertheless, the problem target is different than ours since he only intends to offer a solution to the consumer side. In this way, by ignoring the supplier side, aspects such as the electrical market are disregarded and thus, so is the possibility of influencing customers that we offer to the suppliers (as we will see in chapter 4).

Worthy of mention is also the work of Ygge et al. [YGA96, YA96, AYG96, Ygg97, AY97, YA97, Ygg98a, Ygg98b, YA98, YAA+99, Gus99, YA00] , which is seminal in the area of agents and energy management. Specifically, they combine power load management with market-oriented programming. They introduce a hierarchical structure of *HomeBots*, intelligent agents that represent every load in the system and buy the energy in a system of forward non-combinatorial auctions. With only one energy supplier, this approach places all the initiative on the *HomeBots* so the UCs cannot express their preferences for having more or less demand at a certain time. Moreover, as stated previously in section 2.2.1, market-oriented programming is based on the Theory of Competitive Equilibrium (as opposed to Game Theory). This model studies equilibrium conditions in which participants deal only with parameters such as the price but not with possible actions of the others. This is, agents are cooperative and not competitive, feature that we require to design a electricity markets in which bidder agents compete to be chosen to supply auctioneers energy demand.

2.4 Summary

This chapter has given an introduction to the theoretical background of this work: constraint problems, including distributed constraint satisfaction and optimisation problems, game and auction theory, including combinatorial auctions and energy markets, including an introduction to demand-side management.

Next chapter opens the detailed description of the research carried out in this dissertation and puts the overall architecture of the system in place. Moreover, it describes in accurate the multi-agent system underneath, their relationships and interactions, and their functions.

I saw it myself and it is indeed a wonder past words; for if one were to collect together all of the buildings of the Greeks and their most striking works of architecture, they would all clearly be shown to have cost less labor and money than this labyrinth. Yet the temple at Ephesus and that in Samos are surely remarkable. The pyramids, too, were greater than words can tell, and each of them is the equivalent of many of the great works of the Greeks; but the labyrinth surpasses the pyramids also. It has 12 roofed courts, with doors facing one another, 6 to the north and 6 to the south and in a continuous line. There are double sets of chambers in it, some underground and some above, and their number is 3,000; there are 1,500 of each. We ourselves saw the aboveground chambers, for we went through them so we can talk of them, but the underground chambers we can speak of only from hearsay. For the officials of the Egyptians entirely refused to show us these, saying that there were, in them, the coffins of the kings who had built the labyrinth at the beginning and also those of the holy crocodiles. So we speak from hearsay of these underground places; but what we saw aboveground was certainly greater than all human works. The passages through the rooms and the winding goings-in and out through the courts, in their extreme complication, caused us countless marvelings as we went through, from the court into the rooms, and from the rooms into the pillared corridors, and then from these corridors into other rooms again, and from the rooms into other courts afterwards. The roof of the whole is stone, as the walls are, and the walls are full of engraved figures, and each court is set round with pillars of white stone, very exactly fitted. At the corner where the labyrinth ends there is, nearby, a pyramid 240 feet high and engraved with great animals. The road to this is made underground.

Herodotus, *"History"*

Chapter 3

Overall Architecture of the System

As pointed out in the Introduction, in this Dissertation we address the challenge of redesigning an energy market to benefit from the new conditions that deregulation has brought on. Basically, the most important of these possibilities is that customers may choose their supply from among several energy retailers. Moreover, in order to accomplish Demand Side Management's general objectives, we should be able to provide Utility Companies with a tool to predict the upcoming demand.

Further, we can identify two different areas within the whole design. On the one hand, the marketplace in which the energy is traded. On the other, the customer's place (meaning a flat, a house or a factory) including all his electricity consuming devices. Both components are interconnected and interdependent. The customers estimate their electricity demand and then select the most convenient (say cheapest) tariff among all those submitted by a number of Utility Companies, which eventually buy this energy from the corresponding wholesale electricity market. We focus on the retail market and on the individual customer's demand scheduling. Therefore, in the model depicted in Figure 3.1, energy producers (represented by the factory in the upper left corner) and the wholesale market will be addressed as mere black-boxes attached to the Utility Companies.

As we see, there are clearly two different areas, each one with specifical characteristics, but joined by the aforementioned global objectives, listed as follows:

- *Demand Prediction*: Utility companies require a feasible and reliable demand prediction tool in order to be able to cope with sudden demand variations.
- *Cheap Demand*: The clients' demand must be scheduled to be consumed in the cheapest possible times, not at the cost of comfort and freedom (to consume).

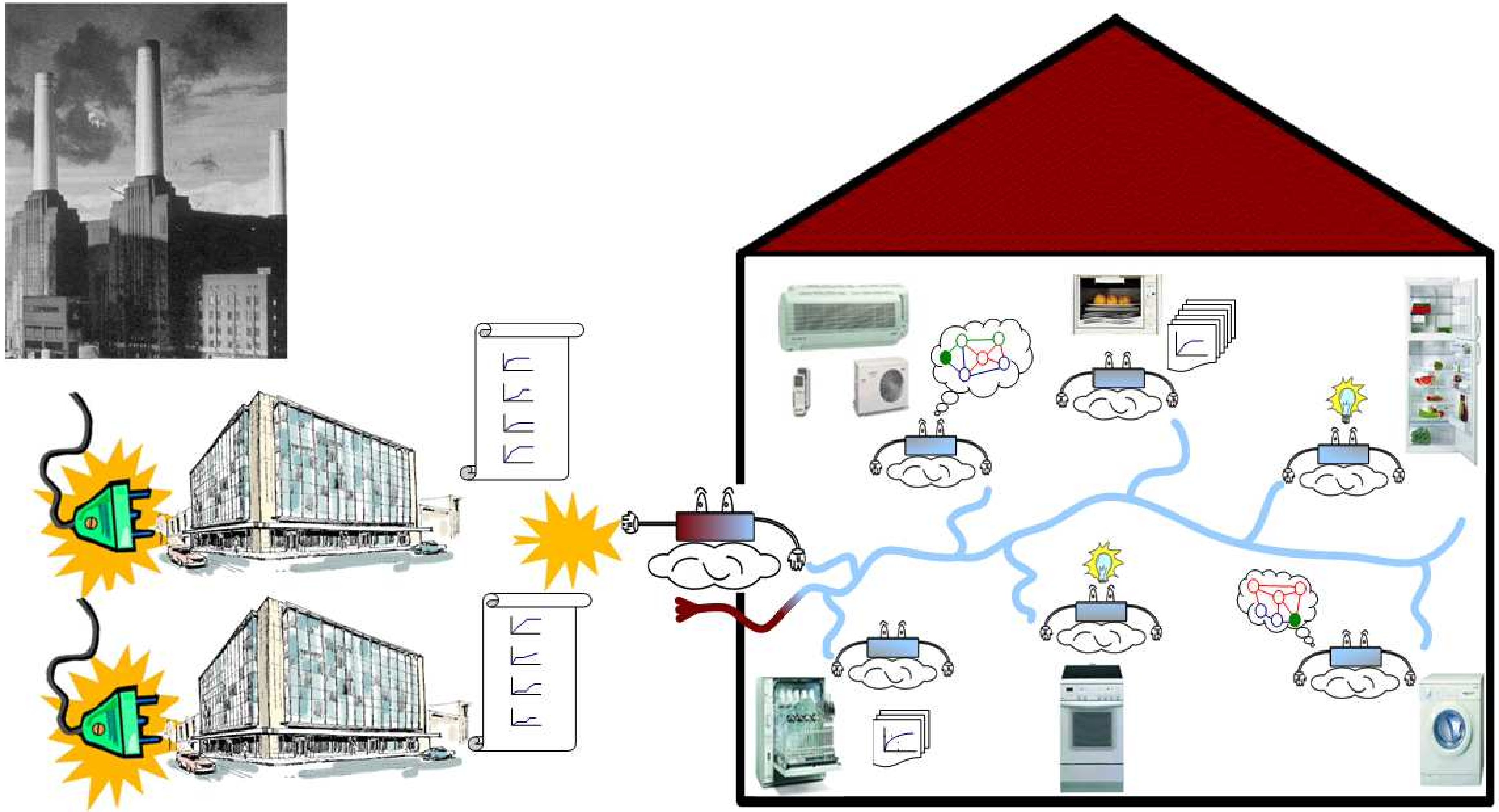


Figure 3.1: **Architecture of the addressed model. –** It includes the scheduling of each single customer's demand (right) and the retail electricity market, where it's supply its agreed (centre). Additionally, the wholesale electricity market (left), where utility companies eventually buy the electricity they sell (not in the focus of this work).

Along with these issues, we may add some further requirements, which are logical consequences from the previous two. For instance:

- *Supply guarantee*: All customers' supply must be met. Therefore, on the one hand, UCs must meet what they offer to supply, and, on the other hand, customers cannot sign up for more energy than the amount, the market can supply.

- *Sensibleness of prices*: Prices shouldn't be too high or too low (thus, *reasonable*[1]). In this way, collusion (both from suppliers or from customers) should be avoided.

- *Blackout prevention*: Energy supply must be assured in case of a sudden demand increment or in peak times (rush hours).

Against this background, we have chosen a multi-agent system (indeed, two) as the mechanism to embody and model the scenario above and help meet the aforesaid goals. Multi-agent technology has a recent but distinguished career in Artificial Intelligence (more accurately as a subdiscipline of it [Les95], distributed artificial intelligence). Its broad remit includes applications in many different areas such as cooperation and coordination [Jen96], scheduling in plant automation [PS02], negotiation [Ram04], diverse varieties of e-commerce and e-business [He04], information filtering [FK96], et cetera[2].

In this way, we have designed the whole architecture as the join of two interconnected multi-agent systems. Each component will be accurately detailed in the following chapters (Chapter 4 for the demand allocation part and Chapter 5 for the demand scheduling one).

Now, let us analyse the nature of the relationships within these components. Agents play a different role in each one of the MAS. In the demand allocation area (see Fig. 3.1, centre) agents *compete* while in the demand scheduling area (see Fig. 3.1, right) agents *cooperate*. Each agent competes in the former in order to be the one chosen to supply energy to each customer (and, at the same time, they try to maximise their revenue when selling this energy). In the latter, agents cooperate to

[1] Note that the definition of a *reasonable* price may be more of a political than a market-based matter. The latter would take into account parameters such as cost of production, cost of delivery, and, eventually, other costs associated to each particular way of generating energy.

[2] This list does neither intend to be complete nor to provide a compilation of paradigmatic applications. See for instance [Woo02] or [OJ96] for a more comprehensive introduction.

achieve an optimal overall [3] consumption schedule. We now shortly introduce the role of each kind of agent.

In the demand allocation part, we have designed a number of simultaneous reverse auctions (one per customer) where each single client sells her energy supply to the highest bid. Therefore, we have:

- *Auctioneer Agents*: They arrange the auction in which the supply of energy for a single customer is allocated. They receive bids from the utility companies and clear the auction.

- *Bidder Agents*: Representing each utility company, these agents try to maximise their revenue by winning the possibility of supplying energy to certain customers (those who organised the auction in which the bidder agent won).

In the demand scheduling part (for each single customer), we have designed a system where devices foresee their upcoming demand and try to adopt it to their counterparts' demand to achieve an optimised overall consumption scheme. As we will see, a device's singular demand is modeled as a set of constraints so the MAS optimises a distributed constraint problem. In this way, agents may be:

- *Device Agents*: These agents model energy consumers' behaviour. They collaborate with each other to achieve the cheapest possible demand schedule. Representing the devices, they can issue a consumption prognosis and, eventually, may reschedule some demand, if this is the best thing to do in order to maximise the overall social welfare.

- *Auctioneer Agents*: They are consulted by the device agents to calculate the cost of a solution (i.e. a certain state defining each device agent's consumption plan), according to the received bids. Hence, they clear the auction for the consulted demand schedule.

In the subsequent chapters we will go on to detail, for each part of the design, the solutions proposed, evaluating them and comparing them to existing state of the art alternatives.

[3] *Overall* has a limited remit here, namely just the group formed by all the consumers of a single client.

3.1 Summary

This chapter has given a brief insight into the solution addressed in this dissertation as a whole. As stated in the introduction, we put forward here a novel electricity market design and a demand optimisation method. In this way, chapter 4 deals with the design of the new market as well as with the algorithms developed to clear the auctions of this special market setting. Then, chapter 5 covers the demand optimisation in the last mile and presents this environment modeled as a constraint optimisation problem and the algorithm developed to solve it optimally.

Sherman sat down before his own telephone and computer terminals. The shouts, the imprecations, the gesticulations, the fucking fear and greed, enveloped him, and he loved it. He was the number one bond salesman, the 'biggest producer', as the phrase went, in the bond trading room of Pierce & Pierce on the fiftieth floor, and he loved the very roar of the storm.
'This Goldman order really fucked things up good!'
'– step up to the fucking plate and –'
'bid 81/2 –'
'I'm away by two thirty-seconds!'
'Somebody's painting you a fucking picture! Can't you see that?'
'I'll take an order and buy'em at 6-plus!'
'Hit the five-year!'
'Sell five!'
'You couldn't do ten?'
'You think this thing continues up?'
'Strip fever in the twenty-year! That's all these jerks keep talking about!'
'– A hundred million July-nineties at the buck'
'– naked short –'
'Jesus Christ, what's going on?'
'I don't fucking believe this!' 'Holy fucking shit!' shouted the Yale men and the Harvard men and the Stanford men. 'Ho-lee fuc-king shit.'

Tom Wolfe, *"The Bonfire of the Vanities"*

Chapter 4

Optimal Allocation of Demand

This part of the work deals with the task of designing a marketplace in which the allocation of customers' demand is traded, maximising both customers' and suppliers' pay-off. We start with an explanation of the requirements and challenges posed by the problem domain and then go on to give an overview on related research in this area. Next, the marketplace (more specifically, an auction) is detailed as well as the algorithms we have developed to clear it in an optimal way. Finally, we evaluate these novel algorithms and present some possible improvements along with a comparison against state of the art counterparts.

4.1 Requirements

As stated before in Chapter 2, electricity retail markets differ from their more traditional counterparts because the good they trade cannot be held in stock or stored. Therefore, retailers must generate the energy on-demand and, thus, work with consumption prognoses. This limitation causes a number of risks, as follows:

1. *Overgeneration*: Since the price of the energy mainly depends on the production cost, generating more electricity than is consumed is not economical. Moreover, the surplus is wasted.

2. *Hypogeneration*: If suppliers cannot match the demand, the lack of energy will cause a power cut (brownout) or, if prolonged, a blackout.

3. *Inconsistent generation*: As pointed out in section 2.3, only gas turbines can react quickly to demand variations. Most of the traditional types of energy generators (e.g. hydroelectric, thermoelectric, nuclear) varying more or less drastically the production on a, say, hourly basis is very expensive (or, simply impossible). Peaky demand can sometimes just not be matched; this is the reason for which energy generators aim at a flat demand [Pen03].

Against this background, retailers require a reliable mechanism to foresee the forthcoming demand. Nevertheless, consumption is typically distributed unevenly along a day. Traditionally, retailers have addressed a two-rate tariff in order to smooth the daily demand profile, as shown in Figure 4.1. That is, suppliers set the price according to the actual demand load so when the demand is high (i.e. during the day) the price is higher and when the demand is low (i.e. at night) the price is also low. This model can easily be refined if, instead of two rates, we offer three, as shown in Figure 4.2, or even an hour-wise tariff, as we will see now.

Assuming the UC has set the hourly rates in accordance to the global demand (i.e. more expensive in times where the demand is higher, cheaper when lower), suppose that, for instance, from 7 to 9 in the morning and from 7 to 9 in the evening are peak hours, from 12am to 3pm and from 10pm to 11pm is a medium-load period and, finally, the rest is low-demand time (off peak).

With this model, an UC improves its preference expression capability. Not only is it clear that it is better to consume at night rather than during the day, but also consuming at certain hours is favoured. This rating scheme can be seen as a communication tool between supplier and customer, where the supplier advises the customer

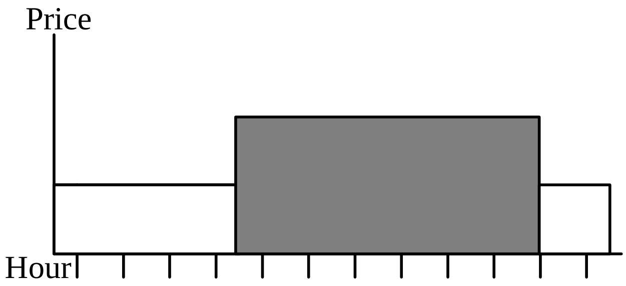


Figure 4.1: **Basic DSM-Tariff –** By providing a dual peak/off-peak tariff, Utility Companies try to influence customer's energy consuming behaviour. These tariffs typically present two different rates: day rate (depicted in dark grey) and night rate (depicted in white).

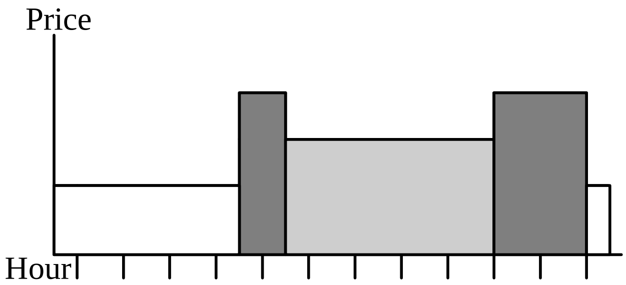


Figure 4.2: **Hour-wise DSM-Tariff –** This models increments the distinction from the former two levels (peak and off-peak) to three levels: peak (depicted in dark grey), medium (depicted in light grey), and off-peak (depicted in white). Further, it allows to set different prices for each hour.

when to place the consumption. From the point of view of the client, the increment of the price at certain hours constitutes an incentive to move the consumption into cheaper hours. The idea here is the same as in tariffs for cellular phones, where it is a usual practice to offer different rates during the day (more expensive) than during the evening and at night. Therefore, if some calls could be postponed to cheaper hours, why not similarly postpone some electricity consuming tasks?

Now, there is still another mechanism that refines the preference elicitation of the UCs, a kind of third dimension in the communication: the discounts. Think of the classical "take 2, pay for 1" or "take 2 and the second one is half price" offers. The customer is rewarded for buying two items of the same good. We can also adopt this idea to our model. As explained in section 2.3, UCs can reward consumption off-peak in order to move some of the demand to less-loaded times. This goal is achieved by setting the price of that time cheaper and by offering discounts of the type "x% for consuming both at 8am (peak) and midnight (off-peak)". In this way, customers willing to win such a discount are *forced* to place some consumption at midnight (in gametheoretical terms, this discount constitutes an *incentive* for that behaviour). As we have seen in Chapter 2, this process can be seen as a *reverse combinatorial auction*. It is reverse, since the client is the one to organise the auction and it is combinatorial because bidding for a bundle of items is valued differently (say cheaper) than bidding for each of the constituent items separately.

Thus, in order to move to a more dynamic environment where the benefits of competition can be more fully realised, we put forward the following requirements for our market design. The arrangement of customers' electricity supply from multiple UCs should be achieved by having contracts that specify the provision of an amount of energy for a certain period of time (say one hour). In this way, suppliers can better set the price of an hour regarding the estimated demand. These contracts should not necessarily be exclusive and, consequently, customers may have agreements with different companies for the same hour if this is the best thing to do. As we have seen in Chapter 2, this features increases the benefits for the auctioneers. Finally, we assume customers auction, on a daily basis, their next 24 hours consumption divided into 24 items (representing one hour each[1]). They subsequently receive bids from the UCs and make their decision for the next 24 hours.

In conclusion, we cope with two challenges: First, we intend to relieve the user of this burden by executing the whole process automatically. Hence, the system must study the existing rates and select tasks that can be rescheduled (postponed or

[1]This is a trade-off between the very static situations of today and the possibility of auctioning on a per minute basis for the coming minute.

anticipated) to optimise the consumption scheme. This issue is addressed in the next chapter (see chapter 5).

Second, we aim at improving and extending the simple market model described above to permit UCs to express more complex aims and, thus, increase their influence on customers. This objective is addressed in the present chapter.

4.2 Related Work

Now, let us briefly refresh part of the Auction Theory introduced in Chapter 2. While combinatorial auctions provide very efficient allocations that can maximise the revenue for the auctioneer, their main drawback is the complexity of the *clearing* process in which buyers and sellers are matched and the quantities of items traded between them are determined. The major problem stems from the fact that clearing combinatorial auctions is NP-hard [FLBS99, San03], which makes combinatorial auctions intractable in the practice [DJ03]. Thus, a significant amount of work has been concentrated on developing strategies to overcome this shortcoming. Unfortunately, most work in this area deals with clearing combinatorial auctions with *atomic propositions* [Nis00, LBST00, Ten00]. This setting implies that bids are either accepted or rejected in their entirety, which may limit the profit for the auctioneer as Dang and Jennings illustrate this case in the following example:

> Consider the case where there are only two bids: $x1$ units of one good at price $p1$ and $x2$ units at price $p2$, and the quantity the auctioneer wants to trade is less than $x1 + x2$ units. In this case, the auctioneer has no choice other than selecting one or other of the two bids. This may prevent the auctioneer from maximising its payoff. For example, the auctioneer may find it more beneficial to accept both bids partially; that is, trade $y1(y1 < x1)$ units with bidder 1 at price $\frac{y1}{x1} * p1$ and trade $y2(y2 < x2)$ units with bidder 2 at price $\frac{y2}{x2} * p2$. Moreover, if the bids are expressed in terms of the correlation between the quantity of items and the price (rather than the simple linear extrapolation above), there will be even more choice for the auctioneer, and, consequently, even more chance of maximising its payoff.

Hence, a more efficient solution is to allow bidding with demand/supply functions [SS01, DJ03], in which bidders submit a function[2] to calculate the cost of the units to be bought or sold. This allows the customer to accept parts of different bids and constitutes a powerful way of expressing complex pricing policies. In our case, production costs can be easily reflected in the supply function and if bids are accepted partially, there may be more than one winner for the same auction and item. This kind of functions was chosen since they can easily approximate any curve.

In our problem domain, this setting enables customers to accept different parts of bids from different bidders so they can get energy simultaneously from several suppliers. Since the transmission and distribution grids are shared and the path followed by the electricity cannot be tracked down, it is impossible to determine the producer of the energy being consumed. Therefore, the hypothesis of customers being simultaneously supplied by several UCs does not pose any technical problems.

Unfortunately, the research on bidding with demand/supply functions has been scarce. Sandholm and Suri [SS01] considered this possibility but only for the single-item case (which is not sufficient, as we will see, since our problems demands a multi-item auction). Moreover, working with a single item implies that the auction is not truly combinatorial [DJ02, DJ03]. In this way, there exist only two algorithms dealing with this auction setting.

In the first one, Dang and Jennings [DJ02] develop the first single-item and multi-item algorithm for multi-unit combinatorial reverse auctions with demand/supply functions. They sacrifice optimality at the cost of running in polynomial time, which is a usual practice in AI (see a compilation of different polynomial algorithms for diverse problems in [Hoc97] or examples in [NWF78, Has92, Met02]). The algorithms are not guaranteed to find the optimal solution, but do produce solutions that are shown to be within a finite bound of the optimal, which sometimes is an acceptable trade-off. Figure 4.3 shows the polynomial multi-unit combinatorial reverse auction clearing algorithm [DJ02]. Still, we would like to have an optimal algorithm for our problem.

In the second one, again Dang and Jennings [DJ03] present another two algorithms for the same environment but this time they are optimal. The strategy they use consists in defining a dominant set containing an increasingly sorted group of single allocations, so they search within this dominant set for the combinations that form the most profitable day allocation. Given m bidders, n items and k, the upper bound on the number of segments of the dominant set, the complexity in a worst case

[2] In case of a forward auction, bidders submit a demand function. In case of a reverse auction, bidders submit a supply function.

POLYNOMIAL MULTI-ITEM CLEARING ALGORITHM WITH SUPPLY FUNCTION BIDDING

Repeat the following steps:

- For all i, j such that $u_i^j > q_j$, set $u_i^j = q_j$. That is, we truncate the supply function to consider only quantities that are not bigger than the demand. This is because to minimise the total price, the auctioneer does not need to buy more units than its demand, as the price functions satisfy the free disposal property.

- Find the bidder a_k such that:

$$\frac{P_k(u_k^1, u_k^2, \ldots, u_k^m)}{u_k^1, +u_k^2, + \ldots + u_k^m} \text{ is minimal,}$$

  then select a_k to provide all its units $(u_k^1, u_k^2, \ldots, u_k^m)$. That is, we consider all the biggest packages offered by the bidders, then choose the packages that offers the lowest average unit price.

  Note that this is not necessarily the package that offers the lowest average in all packages, because a smaller package may have a smaller average unit price.

- Repeat the steps with the new set of bidders A $\setminus a_k$ and demand $q_j^{new} = q_j - u_k^j$.

Figure 4.3: **Polynomial Clearing Algorithm for Combinatorial Auctions with Supply Function Bidding [DJ02] –** For the multi-item case. The solutions generated are within a finite bound of the optimal.

scenario is $O(n \cdot (k+1)^n)$ in the single-item case and $O(mn \cdot (k+1)^{mn})$ in the multi-item. Figure 4.4 shows the optimal multi-item combinatorial reverse auction clearing algorithm [DJ03] (hereafter referred to as *mDJ*, the single-item as *sDJ*).

mDJ: MULTI-ITEM CLEARING ALGORITHM WITH SUPPLY FUNCTION BIDDING

For every tuple $\langle t_i^j \rangle$, $1 \leq i \leq n, 1 \leq j \leq m$ such that t_i^j is a segment on P_i^j:

- For every $j = 1$ to m do:
  - If $\sum_{i=1}^{n} e^j_{i,t_i^j} > q_j$ or $\sum_{i=1}^{n} q^j_{i,t_i^j} > q_j$:
    Continue; // Jump to the next $\langle t_i^j \rangle$ tuple.
  - Sort $\{w_i(\langle t_i^j \rangle)\dot{\pi}^j_{i,t_i^j}\}$ increasingly.
    - If $\sum_{i=1}^{k} e^j_{i,t_i^j} + \sum_{i=k+1}^{n} s^j_{i,t_i^j} > q_j$:

      · Set: $\begin{cases} r_i^j &= e^j_{i,t_i^j}, \forall 1 \leq i \leq k_j - 1 \\ r_i^j &= s^j_{i,t_i^j}, \forall k_j + 1 \leq i \leq n \\ r_i^j &= q_j - \sum_{i=1, 1 \neq k_j}^{n} r_i^j \end{cases}$

      · End k for loop.
- Compare $P(\langle r_i^j \rangle)$ with the price of the best allocation found so far.

Figure 4.4: **The mDJ clearing algorithm [DJ03] –** Optimal clearing algorithm for multi item combinatorial auctions with supply function bidding.

4.3 Market Design

Now, in section 4.1 we concluded that an optimal electricity market format should include simultaneous supply contracts and customers auctioning their upcoming 24 hours demand. These requirements can be best met by structuring the market as a reverse auction, as we have explained. This description is simply the translation of the current electricity model into auction theory terms. Nowadays, clients may choose among several energy suppliers, which previously have advertised their different rates and tariffs. Therefore, clients are already organising reverse auctions in a way.

An exchange (in which multiple buyers and sellers submit their bids and offers to an independent auctioneer that decides the winners (see section 2.2.2)), was rejected because it scales poorly. In practice, the number of customers may be up to tens of thousands, each of which is selling 24 items, and with combinatorial bidding, clearing such an exchange becomes intractable very fast. Unlike exchanges, reverse auctions have the advantage that they may be performed in parallel. This means that the complexity can be divided between the number of customers because instead of one big auction, many *smaller* ones are carried out at the same time. For these reasons, we have designed our system as a series of simultaneous reverse auctions despite the risk of *overbooking*. This is a problem that cannot be underestimated (see the consequences in, for instance, the 2000- 2001 California electricity crisis [Aut01, OS00, HA01] and section 2.3). It refers to the impossibility of UCs of controlling how many customers will accept their bids (and therefore, they cannot predict their total demand). It can be approximated or foreseen with the help of statistical means, but depending on the quality of these measures, the threat of a blackout will be always present. Such a dilemma exists in every market in which a single producer cannot supply all the demand himself. Therefore, a too successful producer could get to a point above which, paradoxically, it is not economical to sell (because it cannot produce so much and, therefore, must buy the difference somewhere else).

Think for instance of an UC starting a price war[3]. In case it suddenly decreases the rates and more customers than expected opt to change to this UC, it will have problems to meet the new forthcoming demand. Less dramatically, a normal UC can also submit an offer that by sheer chance is accepted by more clients than it was supposed to. In other words, when UCs prepare their tariffs, they must also take into account the number of clients they may supply, in addition to the factors mentioned before. Therefore, again, a too successful tariff, ironically, may not be desirable.

From another point of view, carrying out separate auctions in parallel, where duplicated resources are available from each auction, implies the risk of selling the same resource simultaneously in separate auctions at the same time. This is a problem in our overbooking approach and stems from the fact, as stated before, that resource suppliers have uncertainty about the outcome of each single reverse auction. In other words, suppliers place bids *in all auctions simultaneously* as if the auctions were separated entities and their results hadn't influence on each other, which is not the

[3]For an interesting simulation and discussion of price wars with software agents see [KHS98].

case.

In this way, in order to prevent this blackout (worst-case) scenario, we see only two feasible solutions left, detailed as follows.

1. *Airline-like Model*

   In order to minimise the number of unoccupied seats (and thus, the revenue of the flight) within a passenger aeroplane, airlines are allowed to accept a number of bookings higher than the number of the seats available on board, according to figures based upon close statistic evaluation of the matter [ove91]. Then, passengers denied boarding due to overbooking are compensated. Similarly, we may allow UCs to sell electricity beyond their supply capacity. In case of overbooking, we envisage two approaches: Either UCs withdraw (part of) their bids and compensate affected customers[4] or UCs buy the energy they need in a second market. Nevertheless, both pose new game-theoretic issues (such as the influence of second markets and bid-withdrawal possibility in strategies) that require further research. Indeed, the second market together with a non-realistic ceiling price were the main factors which brought about the 2000-2001 California electricity crisis, as described in section 2.3. Therefore, we recommend the solution detailed next.

2. *Regions Model*

   A different approach to this problem is addressed in [HJM04], [HJM05a] and [HJM05b] by Haque, Jennings and Moreau. Specifically, they present a distributed resource allocation protocol that allocates end-to-end network bandwidth by means of using market-based agents that are deployed in a communications network. The agents compete to buy and sell bandwidth resources from auction servers that use combinatorial reverse auctions. Their approach consists of dividing their whole communications network into distinct local regions, in which resources are auctioned exclusively (i.e. only within the region they belong to). This setting provides much needed benefits since communication messages do not have to be broadcast to all auction servers from where resources are needed, but only to the desired ones.

   We can borrow this idea and apply this strategy to our framework in two ways. In the first possibility, customers are concentrated randomly or geographically

[4] This is not an uncommon practice. The (USA) Federal Communications Commission (FCC, http://www.fcc.gov/) [BCJ95, McM94] allows bid withdrawal at a penalty consisting of the maximum of zero or the difference between the value of the withdrawn bid on a license and the highest bid after the withdrawal on that license. See [Por99, Par99] for more information on this subject.

in *heterogeneous* groups, so suppliers can estimate the maximum demand they will offer to each of the groups. In the second modality, customers are concentrated in *homogeneous* groups, according to the priority of their supply. For instance, in a group, all public-service clients (e.g. hospitals, police stations, etc.). In both alternatives, auctions are carried out in steps, in a random order, for the heterogeneous groups case, or in a priority order for the homogeneous groups case and, therefore, the risk of overbooking is minimised, since UCs may improve they demand estimations with real load data.

Coming back to our market design, as combinatorial bidding is permitted, UCs submit their *special* discounts together with the usual hour tariffs. In this case, having 24 hours (or items) means that there may be up to 2^{24} different combinations of discounts. This is obviously a worst-case scenario because, in practice, our experience in the domain indicates that UCs are highly unlikely to issue a different discount for each possible combination. Moreover, we decided that the auctions should be *sealed* (to reveal the least possible information) and *single-round* (to minimise communication and other delays). The auctions also need to be both *multi-item* and *multi-unit*. As each item is the supply of electricity in one hour, there are 24 items to allocate in an auction. In addition, each bidder may not allocate the whole consumption within an hour to a single UC but rather just a portion of it (i.e. some units).

Another important component to set is the price paid by the winner. We do not want to have a first-price auction because it offers incentives for strategic behaviour (i.e. the participants act according to beliefs formed about others' values and types, which does not assure them of maximising their payoff). To circumvent this, we choose a uniform second price for combinatorial auctions (Vickrey-Clarke-Groves) since this has the dominant strategy of bidders bidding their true valuations of the goods (see section 2.2.2). The price paid by the winner is not directly specified in the bid because bidders submit a supply function. Thus, the customer must calculate the energy he wants to consume within a time slot (i.e. the units of that item to be auctioned) and then decide the cheapest combination with the supply functions submitted (i.e. number of units to be allocated with each bidder[5]). Therefore, the bids are accepted partially. To this end, we use the compact notation introduced in [DJ03], where bidders submit for a certain item a piece-wise linear supply function P composed of n linear segments. Each segment l, where $1 \leq l \leq n$, is described by a starting quantity s_l, an ending quantity e_l, a unit price π_l, and a fixed price C_l. Thus, if a customer wants to buy q units of that item from the supplier, it will pay $P_l = \pi_l \cdot q + C_l \quad \text{if } s_l \leq q \leq e_l$. Additionally, bidders submit a correlation function,

[5]This is similar to the Knapsack Problem and is therefore NP-complete [MT90].

ω, which shows the reward or penalty of buying a number of items together (it is this that makes the bidding truly combinatorial). For instance, $\omega_1(A, B) = 0.95$ would mean that if buying x units of items A and y units of B (i.e. consuming x Kw at time A and y Kw at time B), the price paid will have a 5% discount. Thus, if the unit price of item A is p_a and the unit price of item B is p_b, the final price would be $0.95 \cdot ((x \cdot p_a) + (y \cdot p_b))$.

As pointed out before, there is only one *optimal* algorithm to clear such an auction setting. Specifically, the one presented by Dang and Jennings in [DJ03]. Nevertheless, it is inapplicable in our scenario because it scales poorly (as we show in section 4.6). Therefore, with the market described above in place, the next step is to design a clearing algorithm that solves the winner determination problem more efficiently and allows it to be actually applied in realistic contexts.

The remainder of the chapter is devoted to the explanation of the optimal single-item (sPJ) and optimal multi-item clearing algorithm (mPJ) that we have developed for the electricity retail market described above. Furthermore, we analyse their complexity, prove their optimality, and analyse strategies to keep them tractable. First of all, let us introduce some basic definitions that will be used thereafter:

Definition 11 (single allocation) *A single allocation is a set <time-slot t, supplier s, amount q, price p> meaning that s wants to pay p to buy q units of energy to be consumed at time t.*

Definition 12 (allocation) *An allocation is a list containing a number (between one and the number of suppliers) of single allocations that detail the supply of electricity to be provided to the customer at a given time-slot.*

Definition 13 (more profitable allocation) *A more profitable allocation from two alternatives is the one that for a given total demand q, has the lower total price p.*

Definition 14 (optimal allocation) *An optimal allocation is one in which the demand constraint is satisfied and there is no more profitable allocation.*

Definition 15 (optimal day allocation) *An optimal day allocation is a set of 24 optimal allocations, each of which corresponds to a different item (i.e. there is an optimal allocation for each hour).*

The clearing algorithms we present next are related in that the multi-item one is a consecutive and iterative processing of the single-item one (i.e. the result of the multi-item algorithm is obtained by executing the single-item one with different values). Specifically, clearing a single-item case implies finding the optimal allocation for that item, so this enterprise deals only with the supply functions submitted to one item. The multi-item case has a broader remit (an optimal day allocation) and, thus, it also takes into account the relationships between the different items of the optimal allocations (i.e. the correlation functions). Let us first start with the explanation of the single-item case.

4.4 sPJ: Optimal Single Item Clearing Algorithm

Clearing a single-item algorithm with piece-wise supply function bids involves determining the amount to be allocated to each submitted bid function. In essence, in each loop the algorithm selects one segment of each supply function (the one corresponding to the already allocated demand) and allocates k units to the segment with the best price (i.e. the lowest price for k units after applying any relevant discount on the amount). The loop is repeated until the demand is satisfied. Note that the value of k is dynamically assigned in each loop to guarantee the optimality of the algorithm. Specifically, it always has the ending quantity value (e_l) of the shortest segment being evaluated at that moment.

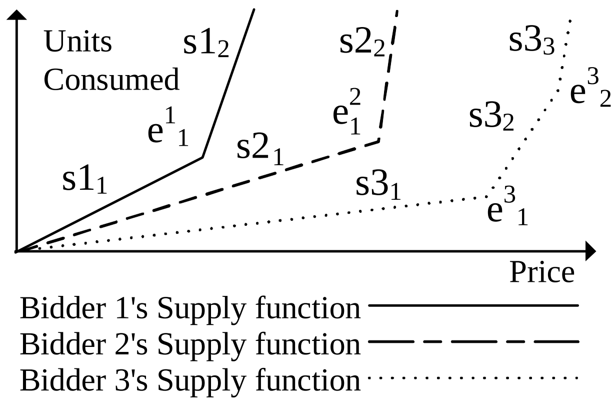


Figure 4.5: **Linear piece-wise supply functions submitted to a single item –** Each bid function is composed of a number of segments S^i_j, where i corresponds to the Bidder and j to the number of the segment, and end points e^j_k, where j shows the number of segments that ends there and k the number of end point.

Let us now illustrate this procedure with the example of Figure 4.5. Assume there are three potential buyers 1, 2, and 3 that submit their supply functions $s1$, $s2$, and $s3$ for a certain item (i.e. the consumption in one hour). In the first loop, the algorithm processes the segments $s1_1$, $s2_1$, and $s3_1$. Since the shortest of the three is segment $s1_1$(i.e. $e_1^1 < e_1^2 < e_1^3$), $k = e_1^1$ and the algorithm compares $s1_1(e_1^1)$, $s2_1(e_1^1)$, and $s3_2(e_1^1)$. Suppose the price of $s3_1(e_1^1)$ is less than the price of $s1_1(e_1^1)$ and $s2_1(e_1^1)$; then, it selects $s3_1$ to supply these first e_1^1 units. In the second loop, the algorithm processes the segments $s1_1$, $s2_1$, and $s3_1$(but starting from e_1^1) and gives k the value of $e_1^3 - e_1^1$ because it is less than e_1^1 and e_1^2. Then, it compares $s1_1(e_1^3 - e_1^1)$, $s2_1(e_1^3 - e_1^1)$, and $s3_1(e_1^1)$, and so on. The algorithm continues until the amount of allocated units is equal to the demand.

As we can see, the algorithm evaluates one function per bidder in each step so it has a complexity $O(m)$ per loop, where m is the number of bidders. As the loop is repeated k times, where k is the number of segments of the function with the highest number of them, the overall complexity is $O(km)$. A safe way to reach an optimal allocation is to select for each unit the segment that offers the best price (i.e. $k = 1$). However, it is not necessary to repeat the process for each single unit since price and discount are constant in each segment. So, as long as the segments evaluated in each loop are the same (unit price and fixed price remain unchanged), the winner will also be the same. Thus, in each loop it is only necessary to compare the price of allocating the lowest ending quantity of the segments being processed, repeating this process until the demand is satisfied. Therefore, sPJ (detailed in Figure 4.6) always finds the most profitable optimal allocation.

4.5 mPJ: Optimal Multi Item Clearing Algorithm

This algorithm, detailed in Figure 4.7, is more complex since it cannot simply be generalised from the single-item one. If there were no correlations, it would be sufficient to run the sPJ case once for each item. However, the existence of correlations poses the problem of the inconsistent application of discounts. First, if a supplier bids for two items and offers a reduction if both bids get accepted, no reduction should be applied if only one of them succeeds. Second, functions become different after applying a discount. For example, assume P_l is a piece-wise supply function for the item l and it is included in the correlation $\omega(l, ...) = x$. Then, P_l' is the new supply function with the value $P_l' = xP_l$. Thus, the optimal allocation of a set of functions in which P_l is included may not be the same as the one in which everything else is the same but with P_l' instead of P_l.

sPJ: SINGLE-ITEM CLEARING ALGORITHM WITH SUPPLY FUNCTION BIDDING

Input: m supply functions f and $demand$.

• Pre-loop: initialise needed variables: $allocated$ to keep the total allocated demand, the list $allocation$ showing the allocated demand per bidder, and the temporal storage variable k.

• Loop: in each loop, until the demand is satisfied, select the segment with the lowest gradient and allocate the minimum ending quantity units.

while *(allocated < demand)* ***do***
  k = select the minimum ending quantity
  if *(demand - allocated < k)* ***then***
    winner = select the minimum f_m(k)
    allocated += k
    allocation[winner] += k
  else
    winner = select the f_m with lowest gradient
    allocated += demand - allocated
    allocation[winner] += demand - allocated

Output: $allocation$, the variable detailing the amount allocated to each bidder.

Figure 4.6: **The sPJ clearing algorithm.** – Optimal clearing algorithm for single item combinatorial auctions with supply function bidding.

In this way, mPJ must process all possible combinations of discounted and non-discounted functions and check that discounts are applied consistently. To this end, we use a brute-force strategy for identifying all the possibilities. Here, all possible bids from each bidder are combined with all possible bids from the rest of the bidders. However, it is not necessary to evaluate all the combinations since some of them are repeated. For instance, Table 4.1 shows an auction with two suppliers (1 and 2) and two items (a and b). In this case, there is one possible correlation for each bidder, $\omega^1(a,b) = x$ and $\omega^2(a,b) = y$. Thus, clearing the multi-item case implies evaluating the combinations where supplier 1 and 2 bid normally for item a (so the single-item clearing algorithm is run with supply functions $P_a^1 - P_a^2$); supplier 1 bids for item a and b with discount and supplier 2 bids normally for item b (so the single-item mPJ: MULTI-ITEM CLEARING ALGORITHM WITH SUPPLY FUNCTION BIDDING

Input: j supply functions s_j, j correlation functions ω_j and demand q_i for each item i.

• Pre-loop: Initialise variable *day-set* to keep the optimal allocation for each item, *item-set* to keep a group of supply functions to be evaluated by the single-item clearing algorithm, $all - item - sets$ to keep already processed sets of supply functions, and a boolean variable ok.

• Loop: For each item calculate the optimal allocation of a possible set of supply functions and then check whether the selected discounts are applicable.

Do
  foreach *item i*
    foreach *supplier* s_j
      add next s_j^i *to item-set*
    if *item-set not in all-item-set* ***then***
      optimal-allocation = single_item_algorithm(item-set)
      store item-set in all-item-sets
    add optimal-allocation to day-set
  ok = check constraints (day-set, ω_j*).*
  if *ok* ***then*** *compare day-set with best so far*
  until *all the combinations are explored*

Output: *day-set*, a set of i optimal allocations (one for each item) with the lowest total price.

Figure 4.7: **The mPJ clearing algorithm –** Optimal clearing algorithm for multi item combinatorial auctions with supply function bidding.

	P_a^2	yP_a^2 yP_b^2	P_b^2	-
P_a^1	$P_a^1 - P_a^2$	$P_a^1 - yP_a^2$ $\boldsymbol{yP_b^2}$	$\boldsymbol{P_a^1}$ $\boldsymbol{P_b^2}$	$\boldsymbol{P_a^1}$
xP_a^1 xP_b^1	$xP_a^1 - P_a^2$ $\boldsymbol{xP_b^1}$	$xP_a^1 - yP_a^2$ $xP_b^1 - yP_b^2$	$\boldsymbol{xP_a^1}$ $xP_b^1 - P_b^2$	$\boldsymbol{xP_a^1}$ $\boldsymbol{xP_b^1}$
P_b^1	$\boldsymbol{P_a^2}$ $\boldsymbol{P_b^1}$	$\boldsymbol{yP_a^2}$ $P_b^1 - yP_b^2$	$P_b^1 - P_b^2$	$\boldsymbol{P_b^1}$
-	$\boldsymbol{P_a^2}$	$\boldsymbol{yP_a^2}$ $\boldsymbol{yP_b^2}$	$\boldsymbol{P_b^2}$	-

Table 4.1: **Enumeration of the single-item evaluations with two items and two bidders –** The repeated combinations are written in bold.

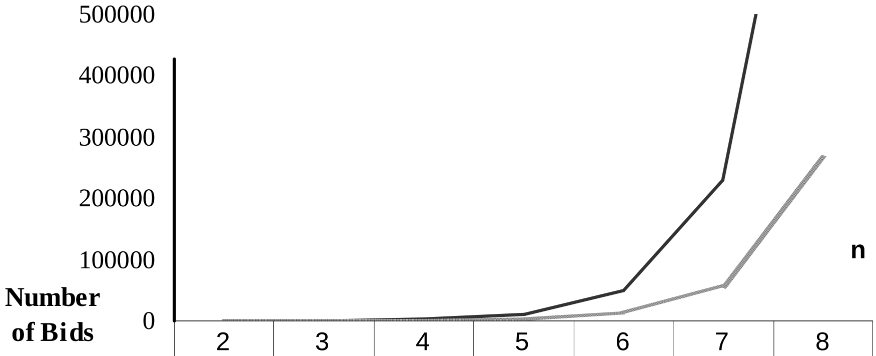
algorithm clears item a with supply function xP_a^1 and item b with xP_b^1 and P_b^2), and so on.

This brute-force strategy evaluates all possible bid combinations (without repeating some of them) and, therefore, it always finds the most profitable optimal day allocation. However, it also scales poorly. First, the number of possible combinations depends on the number of items. In our case, with 24 items, there are 2^{24} different combinations. Second, it also rises exponentially as the number of bidders grows: with n items, and two bidders, 2^{2n}; with three bidders 2^{3n}, and so on. In the extreme situation with two bidders submitting a different supply function for each one of the 24 items and 2^{24} correlations, there are $2 \cdot 2^{2 \cdot 24}$ possible combinations. This is, $n \cdot (2^n)^m$, where n is the number of items and m the number of bidders. For instance, in the example of Table 1, there are $2 \cdot (2^2)^2 = 32$ possible combinations, but half of the combinations do not need to be re-calculated (in bold format in Table 1). Thus, if bidders bid for all items and submit all possible correlations, the number of times that the multi-item algorithm clears the single-item one is $n \cdot (2^n - 2^{n-1})^m = n \cdot (2^{n-1})^m$. Therefore, the complexity is $O(kmn \cdot 2^{(n-1) \cdot m})$, where n is the number of items, m the number of suppliers and k the number of segments of the supply function with more segments. Note, however, that this is a pathological worst-case scenario, which is highly unlikely to happen in practice. Furthermore, as we discuss below, it can be mitigated against by constraining the agent's bidding behaviours.

4.6 Evaluation

In this section we present the results of comparing the performance of our mPJ algorithm with the only other optimal algorithm for this class of problem. Specifically, our benchmarks are the algorithm mDJ presented by Dang and Jennings in [DJ03] (described in more detail in section 4.2, and referred here as "sDJ" for the single-item one and "mDJ" for multi-item).

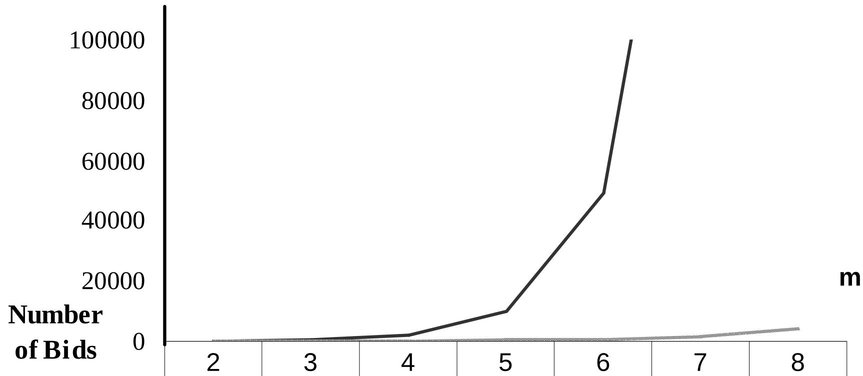
The comparison shown in Figure 4.8 details how the complexity (defined in terms of X, the number of bids) scales when the number of items n increases for a constant number of bidders m. As can be seen, mDJ soon becomes intractable (i.e. prohibitively high complexity), and mPJ scales better.



	2	3	4	5	6	7	8
mDJ	64	384	2048	10240	49152	229376	1E+06
mPJ	16	96	512	2560	12288	57344	262144

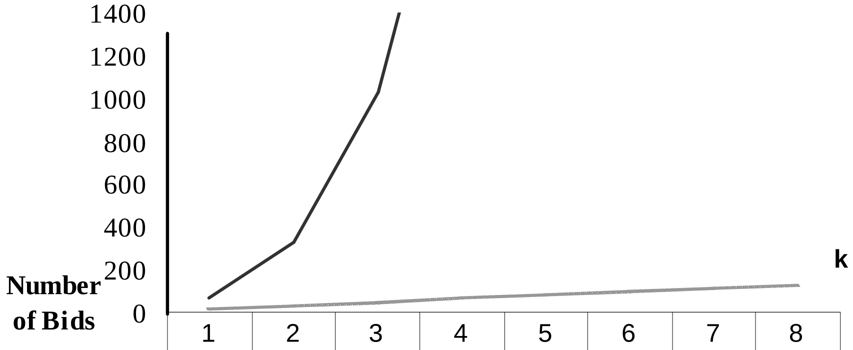
Figure 4.8: **Complexity evolution with n increasing and m steady ($m = 2$).** – The dimensions of the comparison are the number of bids to clear in the x axis, and the number of items (n) in the y axis.

Figure 4.9 tests how the algorithms react to the increment of m (bidders) when n (items) remains steady. Again, mDJ becomes intractable as soon as it did in Fig. 4.8, whereas mPJ presents a significantly better performance profile. The main reason for this behaviour is the sensitivity of mDJ to the increment of both n and m (while mPJ is only sensitive to the increment of n, as seen in Fig. 4.8). For mDJ, a larger number of items and clients means a larger number of single-allocations to form the set from which the allocations will be formed. Whereas for mPJ, more clients means more correlations to clear, but half of which need not be processed since they are repeated.



	2	3	4	5	6	7	8
mDJ	64	384	2048	10240	49152	229376	1E+06
mPJ	16	48	128	320	768	1792	4096

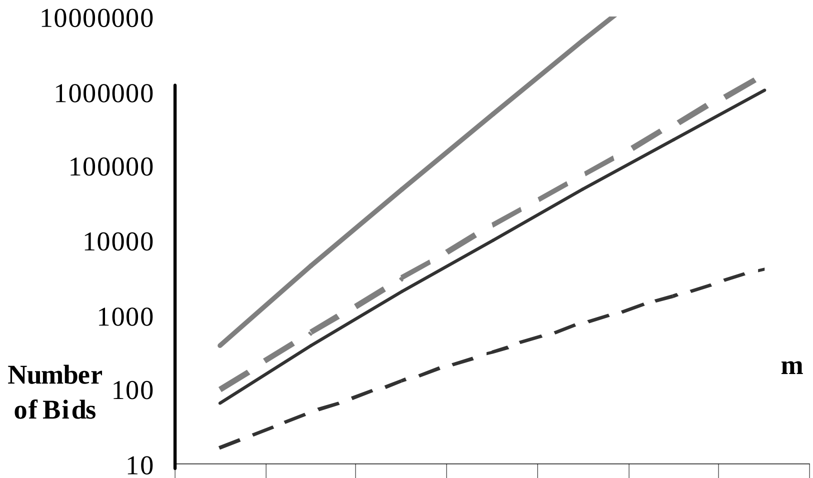
Figure 4.9: **Complexity evolution with m increasing and n steady ($n = 2$).** – The dimensions of the comparison are the number of bids to clear in the x axis, and the number of bidders (m) in the y axis.



	1	2	3	4	5	6	7	8
mDJ	64	324	1024	2500	5184	9604	16384	26244
mPJ	16	32	48	64	80	96	112	128

Figure 4.10: **Complexity evolution with n and m steady and k increasing ($n, m = 2$)** – The dimensions of the comparison are the number of bids to clear in the x axis, and k in the y axis.

Similarly, Figure 4.10 illustrates the behaviour of the algorithms when both n (items) and m (bidders) increase. Again, mDJ performs worse than the others. Its $n = 2$ series is almost equivalent to the $n = 3$ of our multi-item algorithm.



	2	3	4	5	6	7	8
mDJ n=2	64	384	2048	10240	49152	229376	1E+06
mDJ n=3	384	4608	49152	491520	5E+06	4E+07	4E+08
mPJ n=2	16	48	128	320	768	1792	4096
mPJ n=3	96	576	3072	15360	73728	344064	2E+06

Figure 4.11: **Complexity evolution with n and m increasing.** – The dimensions of the comparison are the number of bids to clear in the x axis, and the number of bidders (m) in the y axis.

Finally, Figure 4.11 depicts the dependence of each algorithm on k, the number of units allocated in each iteration of the single-item algorithm. In this dimension, mPJ performs again well. For mDJ, increasing k implies increasing the number of single allocations that may be combined with each other (therefore the algorithm grows exponentially with k as the base). In contrast, for mPJ increasing k just implies that the single-item algorithm is going to process more steps (therefore the algorithm grows linearly with k as the factor).

4.7 Improvements

In order to prevent from occurring such scenarios that could be seen as pathological, it is possible to constrain the choice of possible discounts so bidders can submit only a certain number of correlations. This type of restriction has already been successfully applied to atomic propositions bidding [BCG03], where when limiting the allowable combinations to tree structures or sequential combinations, the NP-hard winner determination problem can be solved in polynomial time [San02]. In a similar vein, mPJ can also take advantage of such an approach. Specifically, we can constrain the number of correlations to a value c. Thus, bidder 1 can issue, for instance, the following: $\omega_1^1(n_1, n_2 \ldots n_i)$, $\omega_2^1(n_1, n_2 \ldots n_i)$ $\ldots \omega_c^1(n_1, n_2 \ldots n_i)$ where i is the number of items included in each discount (for the sake of simplicity, let us suppose it is a fixed number less than n, the number of items, but big enough to allow the bidder to be sufficient flexible in its offering).

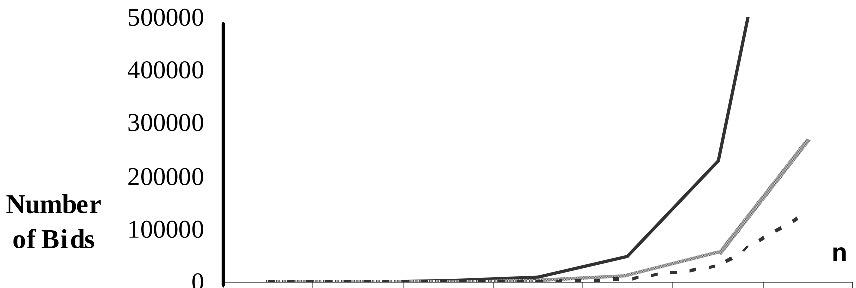
In this way, the single-item algorithm sPJ will be executed, with m bidders, $i \cdot c^m$ times (again, supposing that i is fixed) and the complexity of the mPJ algorithm will drop to $O(ki \cdot c^m)$. Unfortunately, in this case, the mPJ algorithm cannot skip evaluating half of the combinations (as in section 4.7). With this constrained discount choice, the reduction depends much more on the specific discount combinations chosen. For instance, if the combinations include many items (i.e. i is bigger), the single-item algorithm will be executed more often than if the combinations only include two items each. In short, there is no way to accurately determine it *a priori*. Similarly, restricting the available amounts assigned to the discount increases the number of repeated combinations. Thus, if a supplier offers the same reduction for accepting two different items (e.g. $\omega(a, b)$ = $\omega(c, d)$), the number of repeated combinations would increase further and the complexity would continue decreasing.

For the comparison shown next, we have set the maximum number of bids to be issued as half of the maximum possible ($c = 2^{n-1}$) and the maximum number of items included in a correlation as the number of items ($i = n$). For instance, in figure 4.12, we present the reaction of the three algorithms to the increment of the number of items n for a constant number of bidders m.

Here, the constrained variant presents the best profile for our purposes. This would have been even clearer if we had not set the value of c and i depending on the number of items n (as detailed above). With a fixed c and i, the constrained variant would had presented a flat line, whereas mDJ and mPJ would had grown exponentially because in contrast to mDJ and mPJ, the constrained variant does not depend directly on the number of items being auctioned.

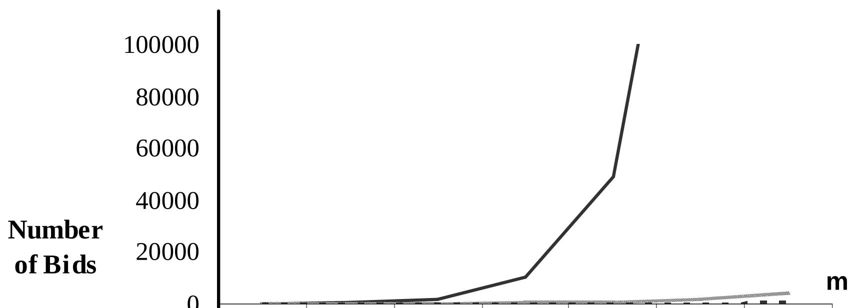
Figure 4.13 tests how the algorithms react to the increment of m (bidders) when n (items) remains steady. The results of the constrained variant are better than those from mPJ. Again, mDJ becomes intractable as soon as it did in Fig. 4.12.

Figure 4.14 presents the reaction of the algorithms to the increment of both n (items) and m (bidders). As we would expect, the best results are again achieved by the constrained variant.



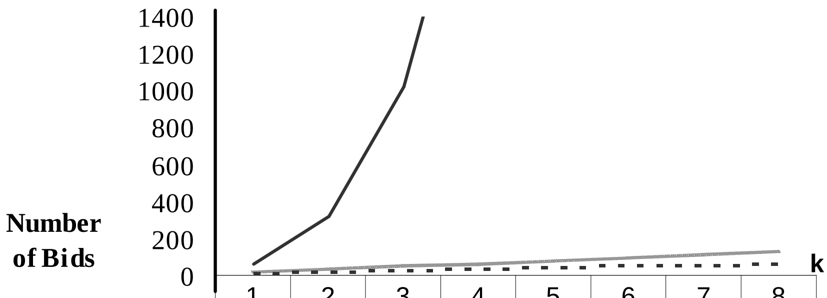
	2	3	4	5	6	7	8
mDJ	64	384	2048	10240	49152	229376	1E+06
mPJ	16	96	512	2560	12288	57344	262144
Constr. bid	8	48	256	1280	6144	28672	131072

Figure 4.12: **Complexity evolution with n increasing and m steady ($m = 2$) –** The comparison are the number of bids to clear in the x axis, and the number of items (n) in the y axis.



	2	3	4	5	6	7	8
mDJ	64	384	2048	10240	49152	22937	1E+06
mPJ	16	48	128	320	768	1792	4096
Constr. bid	8	16	32	64	128	256	512

Figure 4.13: **Complexity evolution with m increasing and n steady ($n = 2$). –** The dimensions of the comparison are the number of bids to clear in the x axis, and the number of bidders (m) in the y axis.

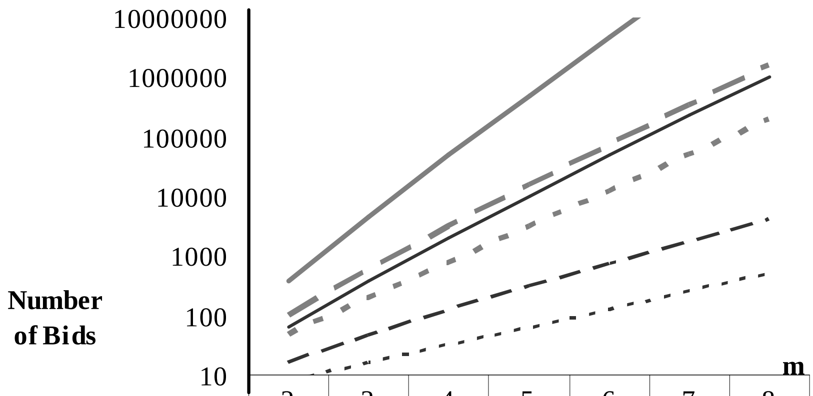


	1	2	3	4	5	6	7	8
mDJ	64	324	1024	2500	5184	9604	16384	26244
mPJ	16	32	48	64	80	96	112	128
Constr. Bid	8	16	24	32	40	48	56	64

Figure 4.14: **Complexity evolution with n and m steady and k increasing ($n, m = 2$) –** The dimensions of the comparison are the number of bids to clear in the x axis, and k in the y axis.

Finally, Figure 4.15 shows the dependence of each algorithm on k, the number of units allocated in each iteration of the single-item algorithm. Again, mDJ is the worst of the three and the constrained variant by far the best. This behaviour is due to the direct dependency of mDJ on k.

Note that the complexity of the constrained variant can be further reduced depending on the values of i and c. With the values we assigned to i and c for these comparisons, it is only m times less complex than mPJ (since $c = 2^{n-1}$, $i = n$ and $O(ki \cdot c^m)$, then the complexity after substitution of c and i is $O(kn \cdot 2^{(n-1)m})$). The genuine advantage of the constrained variant can be found when there are higher values of n and m. Thus, based on our beliefs about the likely operation of the retail energy market some “typical” values might be to have 24 items (e.g. 24 hours) and around 20 bidders (e.g. 20 UCs trying to sell their energy). Therefore, if we set $k = 1$ and restrict the number of possible correlations to 10, each one with 5 items (which experience indicates will provide UCs with enough *persuasive* power), the results are clear: mDJ presents a complexity of $1,498E + 147$, our mPJ $1,429E + 141$ and the constrained variant $5E + 20$. In our opinion, this means the constrained variant is sufficiently close to the optimal to be useful, but is still sufficiently tractable to be practicable.



	2	3	4	5	6	7	8
mDJ n=2	64	384	2048	10240	49152	2E+05	1E+06
mDJ n=3	384	4608	49152	5E+05	5E+06	4E+07	4E+08
mPJ n=2	16	48	128	320	768	1792	4096
mPJ n=3	96	576	3072	15360	73728	3E+05	2E+06
Constr. Bid n=2	8	16	32	64	128	256	512
Constr. Bid n=3	48	192	768	3072	12288	49152	2E+05

Figure 4.15: **Complexity evolution with n and m increasing.** – The dimensions of the comparison are the number of bids to clear in the x axis, and the number of bidders (m) in the y axis.

4.8 Summary

This chapter has introduced a novel electricity market setting in which clients sell their energy demand in reverse combinatorial auctions. Such format allows the auctioneer to maximise its benefits (i.e. obtain the cheapest possible energy) and utility companies get a forecast of the forthcoming energy demand they will have to face.

Still, this model only stipulates that clients receive supply bid functions for each hour and discounts on certain hour combinations, and provides an optimal clearing algorithm to determine the amount to be supplied by each utility company regarding a certain hourly demand. Therefore, there is still a need for a mechanisms that connects the clearing of the auction with the electricity consuming possibilities of each single device within a household. In this way, next chapter details how to do this energy consumption optimisation. Basically, these consuming possibilities are modeled as constraints so the problem becomes a constraint optimisation one, in which the clearing algorithm gives the fitness of each household overall consumption combination.

Yet it must be remembered that what appears to us an extensive, complicated, and yet well ordered institution is the outcome of so many doings and pursuits, carried on by savages, who have no laws or aims or charters definitively laid down. They have no knowledge of the total outline of any of their social structure. They know their own motives, know the purpose of individual actions and the rules which apply to them, but how, out of these, the whole collective institution shapes, this is beyond their mental range. Not even the most intelligent native has any clear idea of the Kula as a big, organised social construction, still less of its sociological function and implications.

B. Malinowski, *"Argonauts of the Western Pacific"*

Chapter 5

Optimal Scheduling of Demand

In this chapter we explain the solution we have developed to optimise the demand scheduling. First we introduce the notion of DSM-able consumers. Basically, a DSM-device can modify its electricity consumption profile according to the system's needs (e.g. in order to adopt the cheapest possible overall consumption plan). We need this concept if we intend to model the demand of a constellation of devices. Then we outline the requirements stemming from the particular setting of our problem domain. Further, we detail how we model it as a distributed constraint optimisation problem. Specifically, we present a novel optimal algorithm to solve this kind of problem and refer to how we tailor existing counterparts so they may operate in the problem domain as well. Finally, we show how this algorithm outperforms the current state of the art methods in this area and conclude with a survey on effective ways to alleviate the network overload that it causes.

5.1 DSM-able consumers

The kind of algorithm we require in our DSM-system needs information about the predetermined future behaviour of each consumer. That is, a demand prognosis. Additionally, each consumer might have alternatives for its energy-consumption plan. That is, it might be able to postpone or anticipate its demand partially or totally. The algorithm is then in charge of choosing the right alternatives so that the overall consumption satisfies the given global goal. The question is where and how does this information come from[1].

One solution would be to simulate all energy-consuming entities in a mathematical model that came up with an exact demand schedule for each single consumer. However, this would create a very rigid system that should be updated after buying a new consumer and a centrality always poses the single-point-of-failure problem. Additionally, the behaviour of certain devices is relatively unpredictable, just think of the heating system of a building being influenced by people, the weather and other stochastic events. Therefore, consumption prognoses in advance become unrealistic since the optimisation algorithm would work with incomplete data. For these reasons, we prefer the method to be distributed, instead of centralised, and on-demand instead of pro-active.

In this distributed fashion, the devices themselves store and maintain their own consumption model. Note that device types may go from simple light bulbs to air-conditioning systems, from very small architecture to very large and complex. The stochastic aspects of the system must be estimated and learned. Thus, sensors for the weather and people, calendars for repeating events, and other sensorial enhancements may increase the “consciousness” of the system. A device might be able to estimate its usage hours in advance, just by realising the facts that it is Monday, it is raining, and a certain person is present.

Furthermore, devices with some autonomy (like washing machines, the heating system, etc.) might also have a choice about when and how to consume energy. If some local task is supposed to be finished by some certain deadline, the device might have some freedom on its local schedule. For instance, instead of just switching on the washing machine, the user could select the preferred due time and the device itself would, in this way, collaborate with the optimisation algorithm to select the best time to start.

According to this ability of helping or collaborating with the scheduling algorithm, we address the following classification of energy consumers:

- *Active*: These devices are the real DSM-able parts in the system. They issue a prognosis about their future consumption and their future alternative behaviour. The are able to chose one of these alternatives to change the overall consumption.

- *Informative*: They only issue a consumption prognosis and lack the possibility

[1]The work discussed in this section was presented in [PPL03].

of actively changing and controlling their consumption. To cope with this, they can and must take part in a DSM environment, but only active devices can actively contribute to the DSM goal.

- *Non-Informative*: The consumption of devices that are not part of the DSM system (no communication-network connection, etc.) can only be measured at a (central) energy meter.

Even non-informative devices must be part of the model and the calculations because they also contribute to the overall energy consumption. We propose two solutions: letting them being represented (individually or in common) by a virtual device or plugging them into a socket that would manage their power supply. We discuss both possibilities later in this chapter.

Both types of DSM-able consumers, active and informative, may be sorted with regard to the kind of prognosis they issue. Moreover, some devices may do precise predictions, while others can only estimate how likely it is that such consumption will happen. Such further classification is the following:

- *Consumers that issue an exact prognosis*: The device knows accurately the amount of energy to be consumed and the certain point of time when it is going to be needed.

- *Consumers that issue a probable prognosis*: The device issues a prognosis, where all the energy-consumption predictions are completed with a probability estimation. More accurately, the consumer does not predict how much energy is going to be needed at what time, but how likely is that a certain amount of energy will be needed at what time. The difference between active and informative devices stems from the amount of data provided. Whereas active consumers provide the different possibilities that they have, informative ones only know the probability and the amount of energy.

The prognosis is obtained from diverse sources such as statistics, learning, Gauss-bell for light switches, et cetera. Thus, the more complex and intelligent the device, the more accurate the prognosis. For instance, a heating system that uses statistics to issue a daily energy consumption prognosis should consider differences between day and night, seasons, weekends or holidays and work-days, etc. On the other hand, there are also different kinds of prognoses, depending on the period to be covered [PDPR97]: long-term and short-term. The behaviour of a DSM-environment may vary regarding what type of prognosis do their devices issue.

Type	Description
Active	Take part in the DSM-process and may regulate their own consumption.
Accurate Prognosis	Probability $= 1$ Example: Task 1: 0,3 kWh at 15:00 (variation $\pm$ 2 min), 30 s Task 2: 0,1 kWh at 15:01 (alternatives at 15.02, 15.06), 16 sec.
Probable Prognosis	Probability < 1 Example: Task 3: 0,05 kWh at 15:00 (variation $\pm$ 10 s), with 89% probability, 1 min Task 4: 0,15 kWh at 16:00 (variation $\pm$ 10 min), with 30% probability, 12 min
Informative	Issue a prognosis but cannot regulate their own consumption.
Accurate Prognosis	Probability $= 1$ Example: Task 5: 0,35 kWh at 15:03, 16 min Task 6: 0,13 kWh at 15:19, 48 s
Probable Prognosis	Probability < 1 Example: Task 7: 0,1 kWh at 15:20 with 30% probability, 1 min Task 8: 0,8 kWh at 15:21 with 30% probability, 3 min

Table 5.1: **Classification of devices involved in a DSM environment** – Active (with accurate or probable prognosis), Informative (with accurate or probable prognosis), and Non-informative.

To summarise, Table 5.1 gives the classification of devices that may be included in a DSM environment.

In theory, there is usually just a small number of really “important” energy consuming appliances which have a consumption high enough to be controlled (e.g. water heating [DC00]). Nevertheless, the real power of these DSM-algorithms is based on their ability to involve all energy consumers in the process. In this way, we must somehow manage to also bring the non-DSM-able appliances to work together if we really want a high-quality outcome. Large office buildings or something similar, may have thousands of independent small consumers like the lighting system or sun blinds. In case they don't implement such features, there is still a way to let them participate in a DSM environment: the *DSM-ification*. When an energy consumer gets DSM-ificated it becomes ready to be included into a DSM system, either as an active or an informative member. Principally, we distinguish three solutions that allow the DSM-ification of an energy consumer:

- *Replacement of the original controller*: Involves the substitution of the original hardware controller of the device by a new DSM-able one. It may be an expensive solution, since the new controller must gather old and DSM-related functionality. Therefore, this modality of DSM-ification is only feasible if the vendor participates in the development of the new controller. Depending on this controller, the consumer will become active or informative. In case it only issues the prognosis, but cannot control the behaviour of the device, it will be an informative one. If the new controller both manages the behaviour of the device and issues a prognosis, it will be an active DSM device. Finally, such replacement is not always possible, since there are some devices that do not allow it due to hardware reasons or just because the internal logic of the devices makes it impossible.

- *Establishing a tight support*: This modality consist of installing embedded sensors that deduce the state of the device. If the device is a simple one, with well-known states (as, for instance, a washing-machine, where it is possible to know the amount of energy needed in every state), it is possible to devise a system that enables the device to be DSM-able. Thus, the sensors would detect the status and then they could issue a prognosis. If the sensors inform a local controller that manages some actuators, they could even influence the behaviour of the device. For instance, some states of a washing machine could be rescheduled or delayed, if needed. Establishing a tight support is cheaper than replacing the controller, but the quality of the DSM behaviour will be lower.

- *Establishing a loose support*: This solution aims at a direct control of the power cord of the device. A controller that participates in the DSM system manages the power that the consumer receives. This modality works with a very small range of simple devices that may be switched off an on without any damage or alteration in their status.

Again, Table 5.2 summarises the different ways of DSM-ification, the benefits they may bring and their problems.

<table>
<tr><td>Controller Replacement</td><td>Consists of: Substitution of the original controller by a DSM-able new one
Advantages: The device becomes DSM active
Disadvantages:
Not always possible
Might be expensive
Only feasible with the help of the vendor</td></tr>
<tr><td>Tight Support</td><td>Consists of: Embedded sensors deduce the aim of the consumer and local controllers manage it
Advantages:
Cheap solution, compared to the previous one
Easy to develop in well-known-state devices
Disadvantages:
Too complicated for devices with too many or dynamical states
Quality of DSM behaviour lower than in previous solution</td></tr>
<tr><td>Loose Support</td><td>Consists of: Management of the power cord
Advantages:
Cheap solution
Easily installable
Disadvantages:
Sometimes has no meaning
Sometimes even not informative
Not always applicable</td></tr>
</table>

Table 5.2: **Summary of the different ways of DSM-ification –** Analysis of the pros and cons of each alternative.

Unfortunately, not all devices can be DSM-ificated by the aforementioned process. In order to still include them in the DSM-calculations, we consider the following solutions:

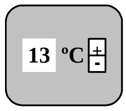
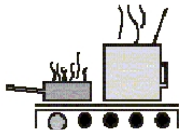
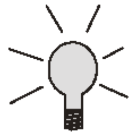
- *DSM-proxy*: The DSM-proxy consists of a socket where the non-DSM devices are plugged in, so they can be switched on or off when needed. It is a coarse DSM-ification with loose support, that maintains all its disadvantages.

- *Virtual device* (VD): A non-DSM consumer (not connected to the DSM communication network), or a group of them, are represented in the DSM process by a so-called virtual device. VDs are software programs (software agents) running on a controller or a computer included in the DSM environment that detects when a certain device is switched on or off with the help of a sensor. If the device is in use (consuming energy) the VD takes part in the DSM negotiation and announces the current consumption of its device. It may even issue a prognosis, in case it is able to learn or use some statistics (just as informative consumers do). The VD might be attached to the global energy meter and it can deduce the usage of the individual appliances via some mathematical analysis and a built-in model [BD01]. In conclusion, VD can be seen as the software version of a DSM-proxy, with the advantage that powerful VDs (those able to use statistics or to learn) may be able to convert their non-DSM-able consumers into informative ones and thus, make them suitable for a DSM process.

Finally, let us have a look at simple example to better illustrate how a DSM-environment works. Think of a DSM-system comprising four different energy consumers: a refrigerator, one vitro-ceramic cooker, a lighting system and a heating system. Table 5.3 illustrates their features their consumption plan at a certain point of time t.

As stated before, there are two additional elements that allow us to include non-DSM-able devices in the DSM process: VDs and DSM-proxies. A number of electrical and portable heating-devices are plugged into a DSM-proxy and a TV virtual device (TV-VD) controls the television set. The TV-VD is a complex software agent that, after a learning process, knows that at time t it is very likely that the TV will stay switched on at least for a further 30 minutes. In case these devices are working, their DSM-relevant data at time t will be as depicted in Table 5.4.

According to this data, all of the devices will need some energy at t (at least probably). The cooker and the lighting system do participate in the DSM process,

 Fridge	 Heating	 Cooker	 Light
DSM-able Exact Prognosis	DSM-able Probable Prognosis	Informative Exact Prognosis	Informative Probable Prognosis

Consumer	**Time**	**kWh**	**Probability**	**Duration**
Fridge	t	0,2	100%	18 minutes
Heating System	t	3	45%	34 minutes
Cooker	t	2	100%	6 minutes
Lighting System	t	0,8	84%	50 minutes

Table 5.3: **Example of a DSM environment** – Each consumers presents a different DSM behaviour (upper table), and a different consumption plan, (lower table).

Consumer	**DSM-ification**	**Time**	**kWh**	**Probability**	**Duration**
TV	Virtual Device	t	0,1	90%	30 minutes
Portable heating	DSM-proxy	t	2	100%	1 second

Table 5.4: **Table example DSM** – Situation of the DSM-ified devices at time t.

since they provide information about the energy they are going to consume. So do the portable heating-devices and the TV, but through their respective means: software (a virtual device) and hardware (a DSM-proxy). Prognosis and information about the energy consumption are, however, not enough to achieve a DSM regulation. After applying a DSM algorithm, only the fridge and the heating system will modify their behavior and this modification is the fact that makes the DSM system feasible. In a critical time or rush hour, even the portable heating devices could be switched off for a while.

5.2 Requirements

Now, let us recall the features of our problem domain as they have been presented so far, in order to list all our requirements. Several UCs submit their bids in the form of supply bid functions. Devices are either DSM-able or get included in the models by means of DSM-proxies or the like. Demand is auctioned with a daily basis (i.e. one day in advance), though consumers with a probable consumption may cause a re-scheduling and, therefore, the re-start of the demand optimisation process. Let us dwell on this subject a bit, with some initial definitions that will help us further.

Definition 16 (single probable prediction) *A single probable prediction is a statement of the form "It is p% likely that the consumption of k kW will take place at time-slot t". Hence, for a given single probable prediction sp, probability p, consumption k and time-slot t,*

$$sp =< p, k, t >$$

Definition 17 (probable prognosis) *A probable prognosis is a set of mutually-exclusive single probable predictions for the same device for a certain time frame (in our case, one day). Hence, for a given probable prognosis pp and n single probable predictions sp_1, sp_2, ..., sp_n,*

$$pp =< sp_1, sp_2, \ldots sp_n >$$

Definition 18 (binding probable prognosis) *A binding probable prognosis is a probable prognosis in which the probability of the constituent single probable predictions sum to 100%. Hence, for a given binding probable prognosis bp and n probable prognoses pp_1, pp_2, ..., pp_n,*

$$\textit{bpp} =< pp_1, p_2, \ldots pp_n > \qquad \textit{where} \sum_{i=1}^{n} pp_i = 100$$

Hence, binding probable prognosis covers all possible cases and, thus, allows us to extract conclusions on the fly. For instance, if device d issues the following prognosis "30% at 9am and 70% at 6pm" and d does not consume at 9am, it will do it *for sure* at 6pm. Therefore, no matter how the system has modeled the situation with the probable consumption (for instance, by dividing the consumption according to the probability) it will be able to re-start the process, this time with more accurate and reliable information (and therefore, the outcome of the algorithm is also likely to be better).

Definition 19 (non-binding probable prognosis) *A non-binding probable prognosis is a probable prognosis in which the probability of the constituent single probable predictions sums to less than 100%. Hence, for a given binding probable prognosis* bp *and probable prognoses* $pp_1, pp_2, \ldots, sp_n$

$$\textit{nbpp} =< pp_1, pp_2, \ldots pp_n > \qquad \textit{where } \sum_{i=1}^{n} pp_i < 100$$

In this case, the system cannot react and take advantage if any of the single probable predictions does *not* come true. It will only be able to do it if, for instance, the first of a number of single probable predictions is realised, since they are mutually-exclusive and, thus, the other will not take place (see def. 5.2).

Anyway, both cases, binding and non-binding, may require a re-scheduling process to adapt the system to continuously updated conditions. Introducing this element also helps us to the uncertainty factor due to human users. In this way, the first requirement of the system, as we summarise in table 5.5, is the need to produce very good quality solutions fast.

Let us focus now on a number of different aspects, this time not related to the devices themselves, but to the marketplace described in the previous Chapter (4). Here it was defined as a "simultaneous reverse combinatorial auction with supply function bidding" (see section 4.3 and [PJ05]). Let us analyse this separately. First, it is an *electrical market*. Second, it is organised as a *simultaneous reverse auction*. Third, the auction is *combinatorial*, and fourth, it allows *supply function bidding*.

The fact that the commodity auctioned is energy consumption is not trivial. It implies that all participants are playing in a common game or, better formulated, that the outcome of the game is obtained as the sum of all players' participation. That is, what device d_1 consumes is summed to what device d_2 consumes, etc. to obtain the global consumption. Therefore, the next requirement we add to the list is the need to use a *global cost function* with two inputs, the global consumption schedule, and the cost of the electricity, and one output, the cost of the overall consumption.

Further, we deal with a reverse auction (so we will receive a number of bids to clear), which is combinatorial, and this makes the clearing process NP-hard, with the following consequences. First, clearing the process determines the cost of the electricity (i.e. clearing the auction *is* the cost function) and, second, due to the hardness of the cost function, the *frequency* with which we call the cost function is a deciding factor. Moreover, the combinatorial nature imposes yet another characteristic: working with *complete* solutions. Thus, we cannot apply bundle discounts until we know the allocation for each single item. That is, our cost function shouldn't examine a partial solution since it won't be able to take into account discounts for consuming at time-slots outside this partial solution. For instance, suppose we have the following demand and bids (represented here as atomic flat-rate bids for the sake of simplicity).

Demand	9:00 am - 5 kW	10:00 am - 4 kW	11:00 am - 3 kW
Bids	Price (kWh)		
UC_1	1.01 cents		
UC_2	0.95 cents		

Note that we have only represented here just part of the day demand. If we clear the auction, the demand supply of 9am, 10am and 11am will go to the cheapest offer, and this is the one by UC_2. Now, if the auction is truly combinatorial, there will exist some bundle discounts. Suppose that UC_1 offers a 10% discount to consuming at 9am *and* 6pm. This discount implies that the consumption supply at 9am would go for UC_1 (0.909 is cheaper than 0.95) but, if we only consider this partial solution from 9am to 11am, how can we know if the discount applies or not?

Finally, allowing supply function bidding provides some new difficulties. Namely, it prevents linearity in the solution space. That is, we cannot sort solutions into any kind of structure (for instance a tree) that would allow us not to evaluate part of them [PJ05]. In other words, there is no possible heuristic that can spare us from having to analyse all the solutions. One solution cannot be supposed to be worse than another (and thus, abandoned) before processing it. For instance, the one that may *seem* worse with its consumption placed in more expensive time slots (or allocated to more expensive suppliers), may take advantage of some discounts that, in the end, make it cheaper.

The specific contribution of supply function bidding is that, as detailed in section 4.4, clearing the auction for a single time-slot (i.e. single item) must be entirely repeated again if one of the constituent function bids is modified because all the resulting allocation could be completely different. In a similar problem domain, but without supply-function bidding and bundle discounts, an algorithm could use any heuristic to estimate the cheapest consumption alternative for a certain task before processing the rest. It would know that all other variants were worse *for sure*. This is not the case in our domain. Again, a task consuming or not at a certain time slot may involve choosing one UC for supplying more or less, or not at all, at another time slot. And this fact cannot be foreseen. Thus, we may draw two conclusions from this last discussion. First, *solution ordering* pre-processing steps do not help here. Second, all possible solutions must be evaluated to assure that the optimal one is found (i.e. the algorithm must be *complete*).

Additionally, we may include another requirement stemming from the concrete application domain of our model. If we open our focus and instead of addressing a single customer's place, we deal with a number of them or, for instance, a whole building or district[2], security and privacy concerns arise that prevents us using centralised methods.

In short, Table 5.5 summarises the requirements imposed by the model[3].

Requirement	**Cause**
Good solution first	Possibility of re-scheduling
Global cost function	Auctioning customer's demand supply
Seldom use of cost function	Np-hardness of clearing process
Complete solutions	Combinatorial nature and supply function bidding
Complete	No linear solution space (Bundle discounts and supply function bidding)
No solution pre-ordering possible	No linear solution space (Bundle discounts and supply function bidding)
Distributed	Security/privacy concerns

Table 5.5: **Model requirements –** These conditions must be fulfilled by the algorithms that aim at optimising the demand.

Unfortunately, there exists no current algorithm able to operate under such conditions, as we detail in the following section.

5.3 Related Work

This section will discuss related work in dCOP algorithms that might help optimising the demand in our model. We will examine state of the art algorithms according to the requirements presented in table 5.5. A more accurate explanation on dCOP algorithms can be found in section 2.1.2.

[2] A coalition of customers may achieve a more powerful position to negotiate better tariffs with providers.

[3] The category "Seldom use of cost function" means that the cost function should be called as less as possible. The cost function must be typically called at least once for each possible solution.

First of all, we may rule out brute-force algorithms that systematically generate and explore all possible solutions. Though complete, they do not fulfil the first condition in Table 5.5: issuing very good quality solutions fast. Thus, we need an algorithm implementing an heuristic search akin to the min-conflict heuristic (see section 2.1).

Adopt, an asynchronous best-first search backtracking dCOP algorithm developed by Modi et al. ([MSTY03, Mod03, MSTM05]) has been on the cutting edge of optimal dCOP algorithms. Similarly to SynchBB [YDIK98] (which marked the state of the art until Adopt's appearance), it does not fulfil a number of the required conditions. First, they both work with partial solutions. As detailed in section 5.2, this use does not make sense in our setting. More over, they operate with local cost functions, whereas we need global cost functions.

Specifically, Adopt uses a pre-processing step to order agents in a tree (or chain), in which constraints play a principal role, since they become into father-child relationships. Namely, trees in Adopt present constraints between ancestors and descendants, but not between siblings. This is clearly not applicable to out problem domain, because all nodes are neighbours. That is, the action (say consumption) of one affects the overall outcome[4].

Lately, Mailler and Lesser [ML04] have presented OptAPO (Optimal Asynchronous Partial Overlay), which apparently outperforms Adopt[5]. Nevertheless, OptAPO is not distributed but semi-distributed (or semi-centralised) and this feature goes against the last requirement of our model (security/privacy concerns), in the same way as centralised algorithms cannot work within the aforesaid conditions.

Against this background, we have two solutions. Either we adapt one of the existing algorithms to work within our problem domain (if possible) or we expressly develop a novel one that fulfils all the requirements. We have tackled both challenges with different success, as we show in the next section.

[4] An additional pre-processing step could help define a number of loosely-coupled neighbourhoods, as shown in section 5.6. Still, the resulting tree would probably be too dense for Adopt's, apart from requiring a pre-processing step for the pre-processing itself.

[5] This claim has created a controversy on the applicability and semantics of the comparison terms (see [DM05]). Similarly, the applicability of OptAPO to real-world scenarios has recently raised some doubts [FB05].

5.4 COBB: Constraint Optimisation by Broadcasting

In this section, we detail the algorithm that we have developed specifically for the problem domain discussed so far, and compare it against state-of-the-art counterparts, tailored to work within the same problem domain. This novel algorithm is called COBB (Constraint Optimisation by Broadcasting, [PJN05]), detailed in Figure 5.1. It is basically a distributed constraint optimisation algorithm where agents broadcast their values and choose the answer proposing the best solution to continue with the algorithm. More specifically, COBB is a synchronous iterative-improvement best-first search algorithm. This is, agents traverse the solution space in a coordinated fashion using an iterative-improvement technique (i.e. only one variable valuation is changed in each cycle) and a best-first search to choose which solution is going to be the basis for the next improvement[6]. Note that COBB always starts with the current solution as a start point for the comparison, so each local change automatically produces a new solution (that can be better or worse than the current one).

The best-first search strategy allows COBB to always choose the best available solution. This tactic, however, could lead it to eventually get stuck in a local maximum. It is in this point where the broadcasting (i.e. including *again* all the other agents in the next step) appears to prevent it (as we illustrate afterwards in the algorithm execution examples, Figures 5.3 and 5.5). Unlike in backtracking algorithms, where the *direction* of the improvement is biased by the hierarchy in which agents are sorted (the child must try all possibilities until she finds no one better and then backtracks to change the father's valuation), the combination of iterative improvement and broadcasting yields much higher freedom. Only one valuation is changed: the one that maximises the global outcome (say social welfare, if we put it as a game). Then, with that new current solution, the whole process starts again.

Essentially, COBB is a recursive algorithm that in each main loop call does the following. First, improve the solution received with local changes. Then, for each possible local change (sorted in descending order) broadcasts the solution and finally, compare each answer (sorted in descending order) with the current best solution and keeps it if it is better.

[6] This is a variant of the classical best-first search [Pea84, RN02], but adapted to operate with complete solutions.

```
CONSTRAINT OPTIMISATION BY BROADCASTING (COBB)

COBB (candidate_solution) {
  new_sol_list = improve (candidate_solution)
  if (new_sol_list != null) {
    foreach new_solution in new_sol_list {
      answers = broadcast (new_solution)
    }
  } else {
    answers = broadcast (candidate_solution)
  }
  sort_desc (answers)
  foreach solution in answers {
    if (fitness(solution) < fitness(best_solution)) {
      best_solution = solution
    } COBB(solution)
  }
}
```

Figure 5.1: **The COBB algorithm –** A novel and optimal synchronous iterative-improvement best-first search algorithm for distributed constraint optimisation problems.

5.5 Evaluation

Let us illustrate this process with an example. Suppose an agent A starts the scheduling process: it takes the current solution and calculates possible improvements by selecting a different value for its own variable (e.g. “if the task can be performed at 9 or 10 am, and the current solution includes its task at 9 am, the agent will asses the whole solution but with its task at 10 am”). Then, it will broadcast the cheapest of both solutions. The other agents will afterwards carry out exactly the same process that agent A has done so far: calculate whether any local change improves the solution and reply. Further, Agent A chooses the cheapest (say best) solution received and restarts the process until no agent responses; in that case it will come back to the second best solution and continue as usual.

The complexity of COBB is $O(2^n \cdot m)$, where n is the number of agents (or tasks) and m the average number of alternatives for a task to be placed. In our system, the minimum allocation unit is the hour and all tasks can be placed at 1 am, 2 am, 3 am

... 12 pm. Thus, in a worst case scenario, where tasks can be placed all over the day, m will be 24 and the complexity $O(2^n \cdot 24)$. Moreover, the algorithm sorts out all possible solutions using a best-first search so more promising ones are processed first. Still, given enough time, all possible solutions are tested and, therefore, the algorithm is complete and it will always find the optimal solution(s).

Finally, let us briefly address the security/privacy issues. In the COBB multi-agent system, participants do know what others are about (or ready) to consume but are unaware of the possibilities they may have to place on this consumption at one or another time slot or to vary their consumption in any way. Moreover, the effort of centralising this information is double. Not only may physically transmitting it be costly but also expressing the consumption possibilities and preferences as well as standardising them maybe very difficult[7].

In order to illustrate the pros and cons of the COBB algorithm and translate it into a framework where it can be compared to some counterparts, we will use a well-known problem modeled as a DCSP: the n-queens problem, which traditionally has been a paradigm of combinatorial problems (see chapter 2). We have chosen it since, as in our problem, there exist constraints between any pair of agents (say queens) and requires a global cost function.

To this extent, we have selected two classical and simple dCSP Algorithms that could be adapted to our domain (by making them work with whole solutions and not partial ones) and are, therefore, potential counterparts.

To this end, Figure 5.2 shows the execution steps followed by one of the algorithms presented in [YH00] and Figure 2.6 (an asynchronous backtracking algorithms), until no queen is threatened. Having four queens, there are subsequently four agents, each one having a variable with values from 1 to 4 (represented as a row in the draughtboard). The start situation for this example is with all variables having same value 1 (i.e. all queens lined up on the left column).

Then, Figure 5.3 presents the execution steps of the COBB algorithm, starting with the same initial situation. In each cycle, COBB first considers local changes and then broadcasts the most convenient of the solutions obtained (with these local changes). In this case, the possible local changes are having the first queen in position 2, 3 or 4 of the first row. Position 2 is still violating a constraint (threatened in the

[7] Think that, similarly as we consider the possibility of having devices with different degrees of DSM-ability, they may also present different technological levels. That is, whereas some of them may be able, for instance, to elicit complex preferences and prognoses, others may be only able to issue very simple predictions. And this is without taking into account potential incompatibilities in languages, standards, and technologies due to strategic brand decisions.

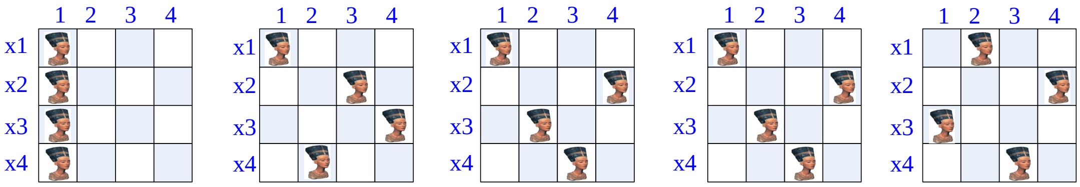


Figure 5.2: **Execution example of the Asynchronous Backtracking algorithm [YH00] –** It needs five cycles to find an optimal solution with this initial state.

diagonal by queen in row 2) and, therefore, it chooses position 3 (4 would be also acceptable) and broadcasts the solution. As this is the one depicted in the middle draughtboard, it is the best of the answers received (the queen of the third row gets value 4). In the next cycle, it does not issue any local change and broadcasts the solution as it is, obtaining a combination where no queen is threatened.

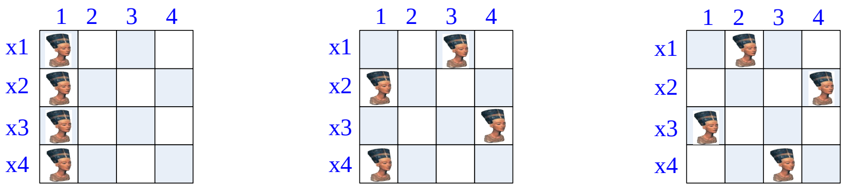


Figure 5.3: **Execution example of the COBB algorithm –** The initial state is the same as in Figure 5.2, and COBB only needs 3 cycles to find an optimal solution.

By comparison, Figure 5.4 illustrates the execution of the asynchronous weak-commitment search ([YH00] and Figure 2.7) which is faster than its counterpart depicted in Figure 5.2. The start situation is this time slightly different, with three constraints violated: queen of row 1 in the same diagonal as queen of row 4, and this one in the same column as queen of row 2. Finally, Figure 5.5 depicts the execution of COBB starting from the same initial situation as in Figure 5.4. The COBB algorithm is faster again.

The execution steps shown in Figures 5.2, 5.3, 5.4, and 5.5 depict just some examples about how COBB's combination of best-first search and broadcasting helps find the solution faster. For the evaluation, we have tested COBB with the distributed n-queens problem varying n from 10 to 50 (as in [YDIK98]). The results are summarised in table 5.6. For each n, we have averaged the results of testing 100 randomly-generated different start situations. One cycle corresponds to a loop of the COBB algorithm or, in case of the other two, a series of agent actions, in which an

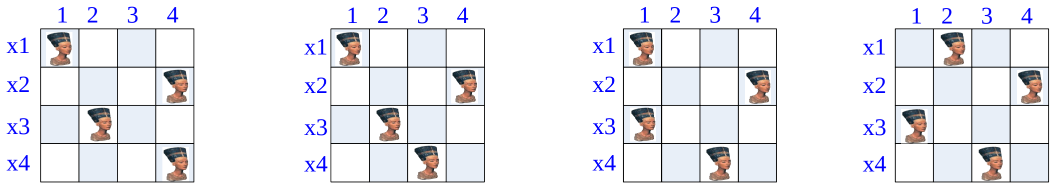


Figure 5.4: **Execution example of the Asynchronous Weak-Commitment [YH00]** – It needs five cycles to find an optimal solution.

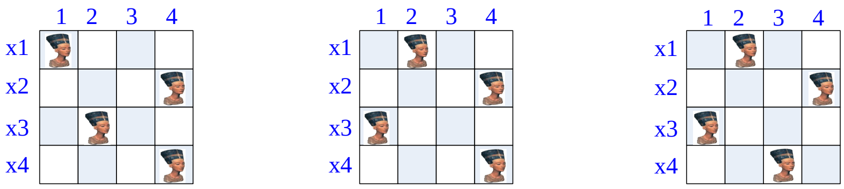


Figure 5.5: **Execution example of the COBB algorithm** – The initial position is the same as in Figure 5.4, and COBB only needs 3 cycles to find an optimal solution.

agent recognises the state of the world, decides its response to that state and communicates it [YDIK98]. We also assume that a message sent at time t arrives at time $t+1$, so we can analyse the performance in terms of cycles needed to find an optimal solution. Finally, following [YDIK98] we have set a time limit for solving the problem (1000 cycles), so $ratio$ expresses the average of runs that successfully completed the problem on time.

	n	10	50
Asynchronous backtraking [YDIK98]	ratio	100%	50%
	cycles	105.4	325.4
Asynchronous weak-commitment [YDIK98]	ratio	100%	56%
	cycles	41.5	59.1
COBB	ratio	100%	100%
	cycles	15.36	35.5

Table 5.6: **Performance comparison in the Distributed N-Queens Problem.** – COBB outperforms Asynchronous Backtracking and Asynchronous Weak-Commitment Search Algorithms in terms of efficiency and speed.

As can be seen, COBB clearly outperforms both algorithms. It is not only faster, but also the most efficient, since it always finds the solution without exceeding the time limit. The reason is that, as illustrated before, the best-first search heuristic helps COBB to quickly come to better solutions and broadcasting enables it to escape from local minima, since all agents participate in each decision. That is, in the other algorithms the view of problem space in each loop is restricted to one agent and its neighbour. In COBB, this restriction disappears with the broadcast, so every agent may contribute to a better solution.

Now, as pointed out before, we had two alternatives: developing a novel algorithm or tailoring any of the existing ones. The latter is what we have done also with Adopt, which was the best of the adaptable existing dCOP algorithms to our knowledge (see section 5.3 and chapter 2 for a description). The new version, which we call *qAdopt* to distinguish it from the original form, works with the same global cost function as COBB (the total number of violated constraints) and it has been adapted to operate with whole solutions (instead of partial ones). The pseudocode is basically the same, so we won't repeat it (see Chapter 2, Figures 2.11 and 2.12).

Nevertheless, using the notion of "cycle" to compare an asynchronous and a synchronous algorithm is not really appropriate (as claimed in [DM05]), at least in our problem setting. It was suited to show how COBB presents higher efficiency than the aforementioned algorithms but it turns out to be inadequate to capture the complexity that our problem domain entails. For this reason, we have compared COBB and qAdopt along the following dimensions: the amount of messages exchanged and the frequency with which the cost function is called (i.e. the frequency with which the combinatorial auction is cleared, each time for a different demand schedule). The latter is a deciding factor in our problem setting, as pointed out before, due to the intrinsic complexity of the clearing process. Hence, maintaining this parameter within a lower range is essential.

We have tested qAdopt and COBB again with the distributed n-queens problem varying n from 5 to 15 queens. The results are summarised in table 5.7 and these are the average of processing 100 randomly-generated different start situations.

Now, in order to more clearly illustrate the different behaviour of both algorithms, Figures 5.6 and 5.7 depict qAdopt's and COBB's profiles and evolution in the same experiments of table 5.7.

As can be seen, with a low number of queens, qAdopt performs better than COBB. Nevertheless, already with 8 queens this tendency is inverted and from that point on, COBB outperforms qAdopt not only in clearing the combinatorial auction less often but also, surprisingly, though using broadcast, in exchanging a lower number of messages. The backtracking mechanism of qAdopt leads to unnecessary

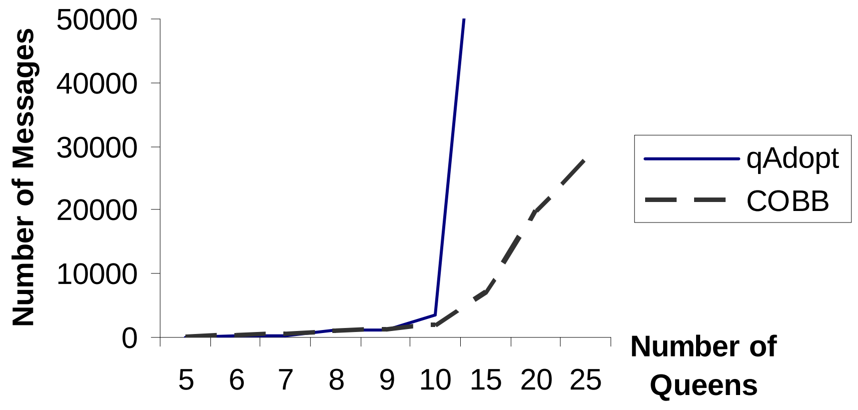


Figure 5.6: **Comparison among qAdopt and COBB in the Distributted N-Queens Problem –** The dimension of the comparison is the number of messages (m).

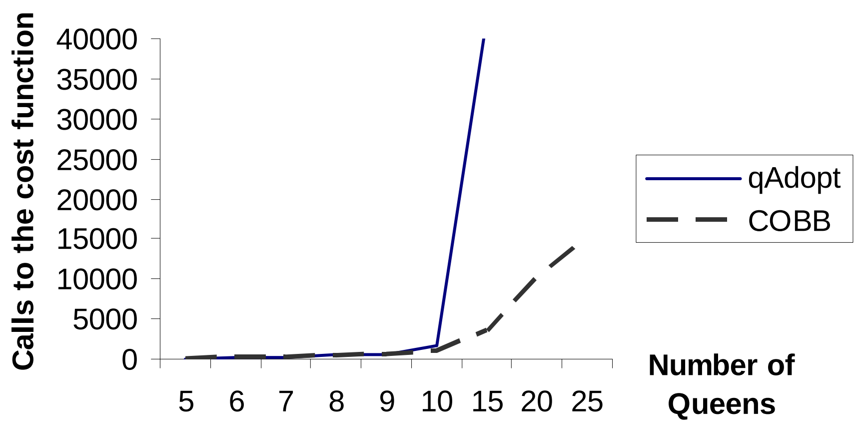


Figure 5.7: **Comparison among qAdopt and COBB in the Distributted N-Queens Problem –** The dimension of the comparison is the number of calls to the cost function (c).

	n	5	6	7	8	9	10	15	20	25
qAdopt	m	63	257	302	1053	1254	3398	85188	- -	- -
	c	35	140	158	555	646	1755	43033		
COBB	m	82	296	460	824	1190	1924	6920	19948	27941
	c	45	161	247	439	629	1012	3579	10230	15255

Table 5.7: **Comparison among qAdopt and COBB in the Distributted N-Queens Problem –** The dimensions of the comparison are the number of messages (m) and calls of the cost function (c).

message overhead to correct previous[8] agent's value assignment. Moreover, for situations with more than 10 queens, qAdopt becomes intractable very fast (as shown in table 5.7, for instance, with 15 queens). That is, qAdopt scales much worse than COBB for our problem domain. Our results are consistent with other performance analyses in different problem domains where Adopt has difficulties when addressing dense constraint graph problems [ML04].

The reason for this different behaviour can be found in the specific general strategy of each algorithm. With the kind of solution space of our problem domain, it is important to explore it in the right direction. Now, this sounds obvious but is exactly what qAdopt does not fulfill. First, allowing agents to change their values asynchronously introduces more flexibility, but also more anarchy. The system disperses its efforts on examining many solutions (and though, potential investigation directions) instead of concentrating on one promising one (as COBB does). Second, the backtrack mechanism locks qAdopt into a rigid structure that prevents good solutions being kept and exploited further and requires an extra communication overhead to pass on to other agents the new solution (and this disadvantage is inherited from Adopt, since it works with partial solutions). Finally, qAdopt does not start to find good quality solutions until the upper and lower bound are well tuned up (so solutions out of that range are abandoned and processed afterwards). This initial tuning process might be too long in case of frequent re-scheduling procedures. When taken together, these reasons and the empirical data shown above make COBB the best algorithm for finding the optimal demand scheduling in our problem domain.

Still, the major objection that can be attributed to COBB is it use of broadcasting. The cost of this choice can be prohibitively high in large systems. In case of a DCSP, the network overhead is alleviated by the fact that the algorithm stops as soon

[8] With "previous" we mean here the antecessor agent in the chain or tree in which Adopt and qAdopt order the agents.

as one solution is found (so the number of messages broadcasted is compensated by the messages that other algorithms interchange in the cycles where COBB is already finished, as happens with qAdopt in Table 5.7). However, this does become a problem when the algorithm has to analyse all the possible solutions. The next section explains how to lessen the impact of this shortcoming.

5.6 Improvements

As stated before, using broadcast in COBB is a double-edged sword. On the one hand, it helps find a very good solution (if not the best) very fast, but, on the other, it requires a high amount of exchanged messages. However, some of these messages need not be sent. Although all agents are virtually neighbours (i.e. can contact each other) they can be in different *neighbourhoods*. With a monolithic tariff this fact means that agents whose consumption alternatives overlap are neighbours (e.g. one consuming at 9 am and 10 am, another at 10 am and 11 am, share 10 am). In our problem domain, the notion of neighbourhood involves more factors. It not only entails the possibility of consuming energy at the same time-slot, but also consuming at a time-slot associated by a discount to any other of the neighbouring agent. For instance, suppose that UC1 offers a 3% discount if consuming at 8 am and 11 am. In case agent A can place its consumption at 8 am and 9 am, its neighbours will be every agent potentially consuming at 8, 9 and 11 am.

There are two reasons for this phenomenon. First, if agent A consumes at 9 and not at 8 am, agents consuming at 11 am will be affected by this fact since the discount could not take place, or viceversa, agents consuming at the same time slot will have to take into account A's consumption to plan their's. In this way, non-neighbouring agents do not need to interchange messages or be part of the broadcast since their local changes will not affect each other. Therefore, determining neighbourhoods will help reduce the network overload by converting broadcast into *selective multicast*. Unfortunately, this reduction can only be exactly evaluated with real data because the composition of neighbourhoods totally depends on the number of discounts, the time slots they include and the number of time slots that each task has on average as alternative to consume energy.

Another, simpler way of reducing network overhead is achieved by keeping the information exchanged to a minimum. COBB is a synchronous iterative improvement algorithm where the current state is broadcasted at the beginning (or is, at least, known by all participants). Then, only one valuation is changed at a time (cycle).

Complete solution	**Bytes**	**One Valuation**	**Bytes**
$Agent_1 - Id$	1		
$Agent_1 - Valuation$	1		
$Agent_2 - Id$	1		
$Agent_2 - Valuation$	1		
$Agent_3 - Id$	1	$Agent_3 - Valuation$	1
$Agent_3 - Valuation$	1		
$Agent_4 - Id$	1		
$Agent_4 - Valuation$	1		
$Agent_5 - Id$	1		
$Agent_5 - Valuation$	1		
Global Cost	1	Global Cost	1
Total	11	Total	2

Table 5.8: **Comparison among sending the complete solution or just a single agent's valuation –** The reduction is 11:2 with n agents and assuming 1 byte for each data type. In this example, agent 3 sends its valuation and the global cost against any other agent sending all the complete solution and the global cost.

Therefore, instead of broadcasting the complete solution together with its cost, agents could just send their new valuation together with the global cost. In this way, they would spare sending $n - 1$ valuations in each cycle per agent (this is, $(n - 1) * n$ valuations).

This time we can present some data to illustrate the effective reduction in network overhead that this measurement brings. Assume that each agent valuation requires for instance 1 byte to express its consumption (to simplify, we will obviate more complex messages as probable prognoses, etc.). The global consumption is similarly expressed in one byte as well each agent's identifier (needed to distinguish each single valuation in the complete solution). Thus, if an agent sends the complete solution and the valuation with, say, 5 agents, the message broadcasted (just the content, excluding TCP/IP Stack headers and the like) will have $(1 + 1) * 5 + 1$ bytes, (ie. 11 bytes). On the contrary, if the agent just sends its valuation and the global cost (its identifier is not needed since it is implicit in the message) the amount transmitted for the content will be 2 bytes. Table 5.8 summarises these calculations.

Similarly, Figure 5.8 illustrates the network overload reduction after applying the aforesaid measure. The data on the number of messages exchanged by COBB has been obtained from the experiments presented in Table 5.7. Whereas sending the complete solution each time scales rapidly, because it depends both on the number of agents and number of messages exchanged, sending only a single valuation increases more slowly because it is independent from the number of agents in the system.

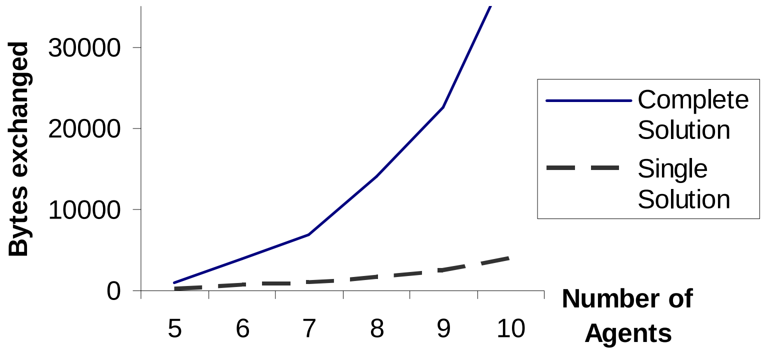


Figure 5.8: **Comparison among sending the complete solution or just a single agent's valuation** – The reduction is 11:3 with n agents and assuming 1 byte for each data type. The y axis shows the amount of bytes exchanged and the x axis the number of agents participating in the system.

5.7 Summary

This chapter has described the method to bring several house appliances to work together and find the cheapest possible consumption profile regarding the supply function and discounts combinations submitted by the utility companies.

After having detailed, tested and evaluated this novel electricity market design integrated within a household demand optimisation system, next chapter accounts and summarises the achievements of this work, discusses them and finalises drawing the avenues of future work.

[Mephistopheles]
Past! A stupid word.
Then, why?
Past, and pure nothing, complete monotony!
What use is this eternal creation!
Creating, to achieve annihilation!
"There, it's past!" What's to read in it?
It's just the same as if it never lived,
Yet chases round in circles, as if it did.
I'd prefer to have the everlasting void.

J.W. von Goethe, *"Faust"*

Chapter 6

Conclusions

In this Chapter, we first summarise the results achieved in this thesis. Then we go on to discuss and explain the results in more detail and, finally, conclude outlining the avenues of future work.

6.1 Summary of Results

This thesis has produced the following results:

- We present, for the first time, a novel **electricity retail market** designed as a system of reverse combinatorial auctions with supply function bidding. This novel market allows customers to increase their profit and provides UCs with a mechanism to influence customers' behaviour.
- We develop new optimal **clearing algorithms** tailored to electricity supply functions and show how they perform better than the existing more general clearing algorithms.
- We present a **taxonomy of electrical devices** according to their degree of participation in a DSM-system and classify the different ways of DSM-ificating a device.

- We introduce the first **DSM-system** of devices modeled as a distributed constraint optimisation problem (dCOP).
- We develop a new optimal **dCOP algorithm** to solve the problem stated above, adapt the state-of-the-art marking dCOP algorithm to the same domain and show that our novel algorithm outperforms it.
- We describe the **integration of the electricity market and customers' private consumption environment** into an architecture that exploits the benefits of deregulated electricity markets to optimally allocate and schedule demand.

Now, let us address the results of this thesis with more detail.

6.2 Discussion

There is a large literature on each of the prevailing research areas that underlie this dissertation: energy markets, distributed artificial intelligence, and game theory. Yet these three strands of work haven't been brought to work together before. Moreover, deregulation has already motivated a number of studies, but this dissertation is the first to fully exploit the possibilities of the new European electricity market. In this way, liberalising the electricity industry offers new opportunities for providers and consumers. In this environment, customers can choose their suppliers to get cheaper energy and suppliers can compete to increase the number of their customers and, subsequently, their profits.

To make this happen in practice, however, efficient electricity markets need to be developed. To this end, traditionally, energy management techniques have presented the two different sides with their own purposes and measures. On one hand, suppliers and retailers aim to smooth the overall energy consumption to avoid sudden peak loads. On the other hand, customers intend to reduce their energy bills without giving up freedom (meaning they can use energy at any time). Our system addresses both needs. It helps to reduce peak loads and to distribute them amongst less-loaded time slots. Specifically, by including off-peak hours in the discounts, UCs reward customers that consume electricity off-peak. Thus, they have an additional tool for energy management besides setting off-peak prices lower than peak ones.

Moreover, the use of combinatorial auctions helps to produce efficient allocations of goods because combinatorial bidding allows the expression of more complex synergies between auctioned items [FLBS99]. Together with the use of supply functions and non-atomic propositions, consumers are able to accept energy from diverse UCs simultaneously, which, in turn, helps them to maximise their benefits.

Against this background, we have presented the first electricity retail market as a system of simultaneous reverse combinatorial auctions with supply-function bidding. Furthermore, we have developed the novel single and multi-item clearing algorithms *sPJ* and *mPJ* that are optimal, as well as a strategy to keep the multi-item algorithm within tractable ranges for the real-world problem we face. The only algorithm that is able to solve this problem optimally, *sDJ* for the single-item case and *mDJ* for the multi-item ([DJ03]) present significantly higher computational complexity even in a worst-case scenario ($O(km)$ in the single-item case and $O(kmn \cdot 2^{(n-1)\cdot m})$ in the multi-item). That is, even if, for instance $k = 1$, our mPJ algorithm is still 2^{n-1} times less complex than mDJ. Moreover, in the constrained bidding variant of mPJ, this difference is even higher. If we again set $k = 1$ and restrict the number of possible correlations to 10, each one with 5 items (which experience indicates will provide UCs with enough *persuasive* power), the results are clear: mDJ presents a complexity of $1,498E + 147$, our mPJ $1,429E + 141$ and the constrained variant $5E + 20$. In our opinion, this means the constrained variant is sufficiently close to the optimal to be useful, but is still sufficiently tractable to be practicable.

Still, it is not enough to find the optimal allocation of demand of a single customer if this demand is not consumed according to the profile. Hence, we need a method that coordinates the customers' devices to find the best consumption schedule with capacity to analyse the submitted bids and find the best consumption profile according to these bids. Thus, we have two interrelated combinatorial problems: finding the best schedule among the possible consumption alternatives of devices and finding the best demand allocation among the submitted supply bids and bundle discounts.

To this end, we first presented a taxonomy of devices regarding their *DSM-ability*. That is, whether they can take part in a DSM-system and if so to what degree. In this way, we distinguish among the following DSM-able device kinds: *Active* that can issue demand prognoses and anticipate or postpone some electricity consuming tasks; *Informative* that can just issue demand prognoses; *Non-Informative* that neither can issue demand prognoses nor adapt their consumption in any way.

In this case, active and informative devices issue their prognoses and active devices also adapt their consumption for the good of all (i.e. in game theoretic terms, to maximise social welfare). If we see the different consumption alternatives of each device as *constraints* and use our clearing algorithms as a cost function, then we have a *constraint optimisation problem*. Further, security and privacy issues, as we have shown, demand that this problem be solved in a distributed manner. Hence, we deal, more accurately, with a *distributed constraint optimisation problem* (dCOP). Following the requirements and properties of our problem domain (specially bids with supply functions and bundle discounts) we have shown that existing dCOP algorithms cannot operate within it.

Against this background, we have developed the novel optimal dCOP algorithm *COBB* (Constraint Optimisation by Broadcasting). This is specially suited to our problem domain. Additionally, we have tailored the state-of-the-art algorithm Adopt, [MSTY03]) to *qAdopt*, and this can operate within our problem domain because it works with complete solutions and a global cost function.

In the absence of data to test them in real-world conditions, we have selected the N-Queens problems to this end, since, similarly to our problem setting, it also demands working with complete solutions and a global cost function, and participants are all neighbours.

In so doing, we have shown how COBB outperforms, in terms of cycles, other algorithms that could be also modified to work within our domain (specifically, Makoto's Asynchronous Backtracking and Asynchronous Weak-Commitment, both in [YH00]). We have selected a new dimension in order to compare them, the frequency of calls to the cost function, because this is more suitable to our domain than the number of cycles. To this extent, COBB has shown that above 8 queens it clearly outperforms qAdopt (which becomes intractable very fast, in our problem domain). Surprisingly, though, allowing using broadcasting, COBB also requires a smaller number of exchanged messages.

Finally, we have detailed possible improvements to both the clearing algorithms and COBB so that the overall system can be made even more applicable to real-world conditions.

6.3 Future Work

There are still some open issues regarding our problem domain. Concerning failure-scenarios and how to keep the demand in secure ranges to avoid blackouts or massive overbooking of the system, more work is needed on which alternative is best, the overbooking-like one (and in this variant, the second market demands more attention due to its potential implications on the strategies of bidders) or the regions-based one.

Furthermore, a number of game-theoretical aspects should be reconsidered. These include the suitability of the Vickrey-Clarke-Groves mechanism (along to the critics of Ausubel and Milgrom in [AM06]) and the possibility of replacing it (for instance with the Ausubel auction [AM06]). In this way, the work of the 2002 Nobel Prize in Economics Kahneman and Tversky [KT73, KT79, KT84] suggests that a human's motivations and decision making do not always follow mathematical (say game theoretical) models and, therefore, it could be worth studying this issue and its influence on both the pricing strategies of suppliers and their real behaviour in our system of simultaneous reverse combinatorial auctions.

In the same way, adopting the regions model as tool for reducing the possibility of overbooking poses new questions regarding UC's pricing strategies because both with heterogeneous and with homogeneous groups the auction would be multi-round.

Moreover, there are some other future work directions that we have already outlined in this thesis (though merely), concerning applicability issues. For instance, the lack of consumption alternatives elicitation procedures and languages or the possible compatibility and standardising conflicts in this area (which can turn a bigger challenge than expected at first sight). This topic is very closely related to the possibility of applying the solution addressed in this Dissertation to a system interconnected by means of a, for instance, a fieldbus system (over power line, for example), where lightweight nodes pose new limitations (see [PMR02]).

Similarly, the establishment of the electricity market format addressed in this work would entail a coordinated effort from all EU-governments (in case of EU-wide implantation) to connect the utility companies with each single consumer. The cost of this process should be added to the one derived of dsm-ificating those single customer's devices and configuring the system to work as is described hereby.

Finally, further work should also focus on developing a simulator to test both the electricity market and the electricity consuming environment with real values. This would lead us to an accurate assessment of the whole system under concrete and realistic conditions.

But mostly, I wanted to tell about her cat. I had kept my promise; I had found him. It took weeks of after-work roaming through those Spanish Harlem streets, and they were many false alarms - flashes of tiger-striped fur that, upon inspection, where not him. But one day, one cold sunshiny Sunday winter afternoon, it was. Flanked by potted plants and framed by clean lace curtains, he was seated in the window of a warm-looking room: I wondered what his name was, for I was certain he had one now, certain he'd arrived somewhere he belonged. African hut or whatever, I hope Holly has, too.

T. Capote, *"Breakfast at Tiffany's"*

Epilogue

While writing my PhD (this report), I've been wondering about which kind of quotation could help me finish the text in a nice way (or less humbly formulated: gracefully, brilliantly). One of the first phrases that came to my mind was "*All's well that ends well*", which would definitively (and effectively) finish the dissertation and, at the same time, add some clever and refined Shakespearian touch.

Throughout my Philosophy studies I've come across Spinoza's work quite often. And I must admit I liked his naivety (and regretted his somehow tragical fate). Thus, it seemed to me a good choice to add the most famous citation of his "(*Ethica Ordine Geometrico Demonstrata*", (*vulgo* simply "*Ethica*"): "*Sed omnia praeclara tam difficilia, quam rara sunt*". "But all excellent things are as difficult as they are rare". Nevertheless, it could be misunderstood as arrogant or even elitist.

And then, lately, I've started to see the whole task differently. Indeed, when I finish it now and submit it, a phase of my life will vanish into memories. This dissertation has been something I've had on my mind during more than four years, present and distant simultaneously, in every act of my life. As a platonic love that someday gets consigned to oblivion by a new, real one. Or not. But, as noted in the prologue, the important is the course, what we have learned and lived in the journey, not the finish. Therefore, I will use Kavafis's poem as farewell to this Dissertation:

As you set out for Ithaka
hope your course is a long one,
full of adventure, full of discovery.
Laistrygonians, Cyclops,
angry Poseidon-don't be afraid of them:
you'll never find things like that on your way
as long as you keep your thoughts raised high,
as long as a rare excitement
stirs your spirit and your body.
Laistrygonians, Cyclops,
wild Poseidon-you won't encounter them
unless you bring them along inside your soul,
unless your soul sets them up in front of you.

Hope your course is a long one.
May there be many summer mornings when,
with what pleasure, what joy,
you enter harbours you're seeing for the first time;
may you stop at Phoenician trading stations
to buy fine things,
mother of pearl and coral, amber and ebony,
sensual perfume of every kind-
as many sensual perfumes as you can;
and may you visit many Egyptian cities
to learn and go on learning from their scholars.

Keep Ithaka always in your mind.
Arriving there is what you're destined for.
But don't hurry the journey at all.
Better if it lasts for years,
so you're old by the time you reach the island,
wealthy with all you've gained on the way,
not expecting Ithaka to make you rich.
Ithaka gave you the marvelous journey.
Without her you wouldn't have set out.
She has nothing left to give you now.

And if you find her poor, Ithaka won't have fooled you.
Wise as you will have become, so full of experience,
you'll have understood by then what these Ithakas mean.

Bibliography

[AM06] L.M. Ausubel and P. Milgrom. The lovely but lonely vickrey auction. In P. Cramton, Y. Shoham, and R. Steinberg, editors, *Combinatorial Auctions*. MIT Press, 2006.

[ARS96] G. Adams, G. Rausser, and L. Simon. Modelling multilateral negotiations: An application to california water policy. *Journal of Economic Behavior and Organization*, 30(1):97–111, 1996.

[ATY00] A. Andersson, M. Tenhunen, and F. Ygge. Integer programming for combinatorial auction winner determination. In *Proceedings of the 4th International Conference on Multiagent Systems*, pages 39–46, 2000.

[Aus04] L.M. Ausubel. An efficient ascending-bid auction for multiple objects. *American Economic Review*, 94(5):1452–1475, 2004.

[Aus06] L.M. Ausubel. An efficient dynamic auction for heterogeneous commodities. *American Economic Review*, 2006.

[Aut01] Various Authors. Manifesto on the california electricity crisis. Available at http://www.haas.berkeley.edu/news/california_electricity_crisis.html, 2001.

[AY97] H. Akkermans and F. Ygge. Smart software as customer assistant in large-scale distributed load management. In *Proceedings of the International Conference and Exhibition on Distribution Automation and Demand Side Management*, 1997.

[AYG96] H. Akkermans, F. Ygge, and R. Gustavsson. Homebots: Intelligent decentralized services for energy management. In J.F. Schreinemakers, editor, *Knowledge Management: Organization,Competence and Methodology*. Ergon Verlag, 1996.

[Bar99] A. Barrett. Autonomy architectures for a constellation of spacecraft. In *Proceedings of the International Symposium on Artificial Intelligence*, pages 291–298, 1999.

[BCG03] A. Babanov, J. Collins, and M. Gini. Scheduling tasks with precedence constraints to solicit desirable bid combinations. In *Proceedings of the 2nd International Conference on Autonomous Agents and Multi-Agent Systems*, pages 345–352, 2003.

[BCJ95] M.M. Bykowsky, R.J. Cull, and J.Ledyard. Mutually destructive bidding: The FCC auction design problem. Social science working paper 916. pasadena: California institute of technology, division of humanities and social sciences, 1995.

[BD01] R. Breed and J. Delport. Non-intrusive load monitoring of residential appliances. In *Proceedings of the 7th Domestic Use of Energy Conference*, 2001.

[BGN03] Y. Bartal, R. Gonen, and N. Nisan. Incentive compatible multi unit combinatorial auctions. In *Proceedings of the 9th conference on Theoretical aspects of rationality and knowledge*, pages 72–87, 2003.

[BH01] C. Boutilier and H.H. Hoos. Bidding languages for combinatorial auctions. In *Proceedings of the 17th International Joint Conference on Artificial Intelligence*, pages 1211–1217, 2001.

[BJW02] S. Bussmann, N.R. Jennings, and M. Wooldridge. Re-use of interaction protocols for decision-oriented applications. In *Proceedings of the 3rd International Workshop on Agent-Oriented Software Engineering*, pages 51–62, 2002.

[BJW04] S. Bussmann, N.R. Jennings, and M. Wooldridge. Multiagent systems for manufacturing control: A design methodology. *Series on Agent Technology, Springer Verlag*, 2004.

[BN02] L. Blumrosen and N. Nisan. Auctions with severely bounded communication. In *Proceedings of the 43rd Annual IEEE Symposium on Foundations of Computer Science*, pages 406–415, 2002.

[BNS03] L. Blumrosen, N. Nisan, and I. Segal. Multi-player and multi-round auctions with severely bounded communication. In *11th Annual European Symposium on Algorithms*, pages 102–113, 2003.

[Boe96] M. Boek. Untersuchungen ueber die Temperaturabhaengigkeit des Stromverbrauches in Oesterreich. Master's thesis, 1996.

[CDK91] Z. Collin, R. Dechter, and S. Katz. On the feasibility of distributed constraint satisfaction. In *Proceedings of the 12th International Joint Conference on Artifcial Intelligence*, pages 318–324, 1991.

[cee03] CEER Council of European Energy Regulators: Directive 2003/54/EC, Directive 2003/55/EC, Commission decision of 11 November 2003. Official Journal of the European Union. Available at http://www.ceer-eu.org/, 2003.

[CGK+01] H. Chalupsky, Y. Gil, C.A. Knoblock, K. Lerman, J. Oh, D.V. Pynadath, T.A. Russ, and M. Tambe. Electric elves: Applying agent technology to support human organizations. In *Proceedings of the Innovative Applications of Artificial Intelligence Conference*, 2001.

[CKL91] S.E. Conry, K. Kuwabara, and V.R. Lesser. Multistage negotiation for distributed constraint satisfaction. *IEEE Transactions on Systems, Man and Cybernetics*, 21(6):1462–1477, 1991.

[CKLM92] S.E. Conry, K. Kuwabara, V.R. Lesser, and R.A. Meyer. Multistage negotiation in distributed constraint satisfaction. *IEEE Transactions on Systems, Man and Cybernetics*, 21(6):1462–1477, November 1992.

[Cla71] E. Clarke. Multipart princing of public goods. *Public Choice*, 2:19–33, 1971.

[Cle96] S.H. Clearwater. Market-based control: A paradigm for distributed resource allocation. *World Scientific*, 1996.

[CLJ96] J. Corera, I. Laresgoiti, and N. R. Jennings. Using archon, part 2: Electricity transportation management. *IEEE Expert*, 11(6):71–79, 1996.

[Com04] Commission of the European Communities. Brussels, 01.03.2004. DG Tren Draft Working Paper. Third benchmarking report on the implementation of the internal electricity and gas market. Available at http://europa.eu.int/comm/energy/, 2004.

[Com05] Commission of the European Communities. Brussels, 15.11.2005. COM(2005) 568 final. Communication From The Commission To The Council And The European Parliament. Report on progress in creating the internal gas and electricity market. Available at http://europa.eu.int/comm/energy/, 2005.

[Cou38] A. Cournot. *Researches into the mathematical principles of the theory of wealth*. Translation to English by N.T. Bacon. The Macmillan Co. (January 1, 1929), 1838.

[Cra06] P. Cramton. Glossary of combinatorial auctions. In P. Cramton, Y. Shoham, and R. Steinberg, editors, *Combinatorial Auctions*. MIT Press, 2006.

[Cro02] R.T. Crow. What works and what does not in restructuring electricity markets. *Business Economics*, (37):41–56, 2002.

[CS03] V. Conitzer and T.W. Sandholm. Computational criticisms of the revelation principle. In *Proceedings of the 5th Workshop on Agent Mediated Electronic Commerce*. LNCS, Springer, 2003.

[CSC+93] R. Caulder, J.E. Smith, A.J. Courtemanche, M.F. Mar, and A.Z. Ceranowicz. Modsaf behavior simulation and control. In *Proceedings of the 3rd Conference on Computer Generated Forces and Behavioral Representation*, pages 347–356, 1993.

[DC00] J. Delport and C. Chen. Hot water load control on campus as part of a virtual power station. In *Proceedings of the 7th Domestic Use of Energy Conference*, pages 11–14, 2000.

[DHKT01] R. Das, J.E. Hanson, J.O. Kephart, and G. Tesauro. Agent-human interactions in the continuous double auction. In *Proceedings of the 17th International Joint Conference on Artificial Intelligence*, pages 1169–1187, 2001.

[DJ02] V.D. Dang and N.R. Jennings. Polynomial algorithms for clearing multi-unit single item and multi-unit combinatorial reverse auctions. In *Proceedings of the 15th Eurepean Conference on Artificial Intelligence*, pages 23–27, Lyon France, 2002.

[DJ03] V.D. Dang and N.R. Jennings. Optimal clearing algorithms for multi-unit single item and multi-unit combinatorial auctions with demand/suppy function bidding. In *Proceedings of the International Workshop on Internet Computing and E-Commerce*, pages 25–30, Pittsburgh PA, 2003.

[DK01] A. Davenport and J. Kalagnanam. Price negotiations for procurement of direct inputs. IBM Research Report RC 22078, 2001.

[DM05] J.P. Davin and P.J. Modi. Impact of problem centralization in distributed constraint optimization algorithms. In *Proceedings of the 4th International Joint Conference on Autonomous Agents and MultiAgent Systems*, pages 1057–1066, 2005.

[DP85] R. Dechter and J. Pearl. Generalized best-first search strategies and the optimality of A*. *Journal of the Association for Computing Machinery*, 32(3):505–536, 1985.

[DPJ03] R.K. Dash, D.C. Parkes, and N.R. Jennings. Computational mechanism design: A call to arms. *IEEE Intelligent Systems*, 18(6):20–47, 2003.

[DTWZ94] A.J. Davenport, E.P.K. Tsang, C.J. Wang, and K. Zhu. GENET: A connectionist architecture for solving constraint satisfaction problems by iterative improvement. In *National Conference on Artificial Intelligence*, pages 325–330, 1994.

[dVV03] S. de Vries and R.V. Vohra. Combinatorial auctions: A survey. *INFORMS Journal on Computing*, (3):284–309, 2003.

[EGKL01] M. Eso, S. Ghosh, J. Kalagnanam, and L. Ladanyi. Bid evaluation in procurement auctions with piece-wise linear supply curves. IBM Research Report RC 22219, 2001.

[EKLL01] M. Eso, J. Kalagnanam, L. Ladanyi, and Y.G. Li. Winner determination in bandwidth exchanges. Presentation at the Infonomics Workshop, Maastricht, The Netherlands, 2001.

[Eme95] R.H. Emerick. *Power plant management*. McGraw-Hill, 1995.

[Far00] P. Faratin. *Automated Service Negotiation Between Autonomous Computational Agents*. PhD thesis, University of London, Queen Mary College, 2000.

[Fas95] D. Faschinger. Analyse der haushaltsstromverbrauchsstrukturen in niederoesterreich ermittlung des einflusses verschiedener einfluparameter. Master's thesis, 1995.

[FB05] P.R. Ferreira and A.L.C. Bazzan. Distributed meeting schedule through cooperative mediation: analysing optapo's performance in a complex scenario. In *Proceedings of the 6th International Workshop on Distributed Constraint Reasoning*, pages 101–113, 2005.

[FK96] I.A. Ferguson and G.J. Karakoulas. Multiagent learning and adaptation in an information filtering market. In Sandip Sen, editor, *Working Notes for the AAAI Symposium on Adaptation, Co-evolution and Learning in Multiagent Systems*, pages 28–32, Stanford University, CA, 1996.

[FL98] D. Fudenberg and D.K. Levine. *The Theory of Learning in Games*. MIT Press, 1998.

[FLBS99] Y. Fujishima, K. Leyton-Brown, and Y. Shoham. Taming the computational complexity of combinatorial auctions: Optimal and approximate approaches. In *Proceedings of the 16th International Joint Conference on Artificial Intelligence*, pages 548–553, 1999.

[Fre89] E.C. Freuder. Partial constraint satisfaction. In *Proceedings of the 11th International Joint Conference on Artificial Intelligence*, pages 278–283, 1989.

[Fri93] D. Friedman. The double auction market institution: A survey. In D. Friedman and J. Rust, editors, *The Double Auction Market: Institutions, Theories and Evidence*, pages 3–25. Addison-Wesley, 1993.

[FT91] D. Fudenberg and J. Tirole. *Game Theory*. MIT Press, 1991.

[FW92] E.C. Freuder and R.J. Wallace. Partial constraint satisfaction. *Artificial Intelligence*, 58(1-3):21–70, 1992.

[FY05] B. Faltings and M. Yokoo. Introduction: Special issue on distributed constraint satisfaction. *Artificial Intelligence*, 16(1-2):1–5, 2005.

[GAO03] GAO - United States General Accounting Office. Electricity Restructuring: 2003 Blackout, Identifies Crisis and Opportunity for the Electricity Sector, Available in http://www.gao.gov/new.items/d04204.pdf, November, 2003.

[Gib73] Allan Gibbard. Manipulation of voting schemes: A general result. *Econometrica*, 41:587–602, 1973.

[GL77] J.R Green and J.J. Laffont. Characterization of satisfactory mechanisms for the revelation of preferences for public goods. *Econometrica*, 45:427–438, 1977.

[GL00] R. Gonen and D. Lehmann. Optimal solutions for multi-unit combinatorial auctions: Branch and bound heuristics. In *Proceedings of 2nd annual ACM Conference on Electronic Commerce*, pages 13–20, 2000.

[Gro73] T. Groves. Incentives in teams. *Econometrica*, 41:617–31, 1973.

[GS93] D.K. Gode and S. Sunder. Allocative efficiency of markets with zero intelligence traders: Market as a partial substitute for individual rationality. *Journal of Political Economy*, 101:119–137, 1993.

[Gus99] R. Gustavsson. Agents with power. *Communications of the ACM*, 42(3):41–47, 1999.

[HA01] R. Haas and H. Auer. Die kalifornische stromkrise. *Energiewirtschaftliche Tagesfragen*, 51(5):280 – 287, 2001.

[Haa95] R. Haas. *Demand-Side Management-Programme in den USA: Historische Entwicklung und aktueller Stand*. VEOE, 1995.

[Haa03] Reinhard Haas. Oeffentliche infrastrukturpolitik: Eine analyse der randbedingungen fr gesellschaftlich optimale strukturen. *Beitraege zur Wirtschaftspolitk, Studie im Auftrag der Kammer fuer Arbeiter und Angestellte fuer Wien*, (14), 2003.

[Has92] R. Hassin. Approximation schemes for the restricted shortest path problem. *Mathematics of Operations Research*, 17(1):36–42, 1992.

[HB91] M. N. Huhns and D. M. Bridgeland. Multiagent truth maintenance. *IEEE Transactions on Systems, Man and Cybernetics*, 21(6):1437–1445, 1991.

[He04] M. He. *Designing Bidding Strategies for Autonomous Trading Agents*. PhD thesis, Electronics and Computer Science, University of Southampton, 2004.

[HHV95] S.P. Hargreaves-Heap and Y. Varoufakis. *Game Theory: a Critical Introduction*. Routledge, 1995.

[HJL03] M. He, N.R. Jennings, and H.F. Leung. On agent-mediated electronic commerce. *IEEE Transactions on Knowledge and Data Engineering*, 15:985–1003, 2003.

[HJM04] N. Haque, N.R. Jennings, and L. Moreau. Resource allocation in communication networks using market-based agents. In M. A. Bramer, editor, *Proceedings of the 24th SGAI International Conference on Innovative Techniques and Applications of Artificial Intelligence*, pages 187–200, 2004.

[HJM05a] N. Haque, N.R. Jennings, and L. Moreau. Resource allocation in communication networks using market-based agents. *International Journal of Knowledge-Based Systems*, 18(4-5):163–170, 2005.

[HJM05b] N. Haque, N.R. Jennings, and L. Moreau. Resource allocation in communication networks using market-based agents. In *Proceedings of the 1st International Workshop on Incentive-Based Computing*, pages 30–39, 2005.

[HNR68] P. Hart, N. Nilsson, and B. Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics*, 4(2):100–107, 1968.

[Hoc97] D. Hochbaum, editor. *Approximation Algorithms for NP-Hard Problems*. PWS Publishing Company, 1997.

[HY97] K. Hirayama and M. Yokoo. Distributed partial constraint satisfaction problem. In *Proceedings of the 3rd International Conference on Principles and Practice of Constraint Programming*, pages 222–236, 1997.

[HY00] K. Hirayama and M. Yokoo. An approach to over-constrained distributed constraint satisfaction problems: Distributed hierarchical constraint satisfaction. In *Proceedings of the 4th International Conference on Multiagent Systems*, 2000.

[IZE94] IZE. Strombasiswissen 124, Drei Stufen fuer den Einsatz von Kraftwerken: Grundlast, Mittellast und Spitzenlast, Frankfurt am Main, S.4, 1994.

[JB03] N.R. Jennings and S. Bussmann. Agent-based control systems. *IEEE Control Systems Magazine*, 23(3):61–74, 2003.

[JCL+96] N.R. Jennings, J.M. Corera, I. Laresgoiti, E.H. Mamdani, F. Perriollat, P. Skarek, and L.Z. Varga. Using ARCHON to develop real-world DAI applications for electricity transport management and particle accelerator control. *IEEE Expert*, 11(6):64–70, 1996.

[Jen96] N.R. Jennings. Coordination techniques for distributed artificial intelligence. In G.M.P. O'Hare and N.R. Jennings, editors, *Foundations of Distributed Artificial Intelligence*, pages 187–210. John Wiley & Sons, 1996.

[Jos03] P.L. Joskow. Electricity sector restructuring and competition: Lessons learned. Cambridge university, available at http://web.mit.edu/ceepr/www/2003-014.pdf, 2003.

[Kar72] R.M. Karp. Reducibility among combinatorial problems. In R.E Miller and J.W. Thatcher, editors, *Complexity of Computer Computations*, pages 85–103. Plenum Press, 1972.

[Kas86] S. Kasif. On the parallel complexity of some constraint satisfaction problems. In *Proceedings of the 5th National Conference on Artificial Intelligence*, pages 349–353, 1986.

[KDMT01] N. Kfir-Dahav, D. Monderer, and M. Tennenholtz. Mechanism design for resource bounded agents, 2001.

[KHS98] J.O. Kephart, J.E. Hanson, and J. Sairamesh. Price-war dynamics in a free-market economy of software agents. In *6th Conference on Artificial Life*, pages 53–62, University of California at Los Angeles, California, 1998.

[Kre90] D.M. Kreps. *A Course in Microeconomic Theory*. Harvester Wheatsheaf, 1990.

[Kri02] V. Krishna. *Auction Theory*. Academic Press, 2002.

[KS00] F. Kelly and R. Steinberg. A combinatorial auction with multiple winners for universal services. *Management Science*, 4(46):586596, 2000.

[KT73] D. Kahneman and A. Tversky. On the psychology of prediction. *Psychological Review*, (80):237–25l, 1973.

[KT79] D. Kahneman and A. Tversky. Prospect theory: An analysis of decisions under risk. *Econometrica*, (47):313–327, 1979.

[KT84] D. Kahneman and A. Tversky. Choices, values and frames. *American Psychologist*, (39):341–350, 1984.

[KTN+99] M. Kitano, S. Todokoro, I. Noda, H. Matsubara, and T. Takahashi. Robocup rescue: Search and rescue in large-scale disaster as domain for autonomous agent research. In *Proceedings of the IEEE International Conference on System, Man, and Cybernetics*, 1999.

[KY99] P.E. Klein and N.E. Young. Approximation algorithms for np-hard optimization problems. In *Algorithms and Theory of Computation Handbook*. CRC Press, 1999.

[LBPS00] K. Leyton-Brown, M. Pearson, and Y. Shoham. Towards a universal test suite for combinatorial auction algorithms. In *Proceedings of the 2nd ACM conference on Electronic commerce*, pages 66–76, 2000.

[LBST00] K. Leyton-Brown, Y. Shoham, and M. Tennenholtz. An algorithm for multi-unit combinatorial auctions. In *Proceedings of the 17th National Conference on Artificial Intelligence*, pages 13–20, Minneapolis MN, 2000.

[LC88] V.R. Lesser and D.D Corkill. The distributed vehicle monitoring testbed: A tool for investigating distributed problem solving networks. In R. Engelmore, editor, *Readings from the AI Magazine*, volume 1-5, pages 69–85. AAAI Publishers, 1988.

[Les95] V.R. Lesser. Multiagent systems: An emerging subdiscipline of ai. *ACM Computer Survey*, 27(3):340–342, 1995.

[LLN01] B. Lehmann, D.J. Lehmann, and N. Nisan. Combinatorial auctions with decreasing marginal utilities. In *Proceedings of the 3rd ACMConference on Electronic Commerce*, 2001.

[LOS99] D.J. Lehmann, L.I. O'Callaghan, and Y. Shoham. Truth revelation in approximately efficient combinatorial auctions. In *Proceedings of the 1st ACM Conference on Electronic Commerce*, pages 96–102, 1999.

[LS95] J. Liu and K. Sycara. Exploiting problem structure for distributed constraint optimization. In V.R. Lesser, editor, *Proceedings of the 1st International Conference on Multi-Agent Systems*, pages 246–254. MIT Press, 1995.

[LS01] K. Larson and T.W. Sandholm. Computationally limited agents in auctions. In *AGENTS-01 Workshop of Agents for B2B*, pages 27–34, 2001.

[Mac92] A.K. Mackworth. Constraint satisfaction. In S.C. Shapiro, editor, *Encyclopedia of Artificial Intelligence*, pages 285–293. John Wiley and Sons, 1992.

[man01] In M. Manninger, K.M. Goeschka, C. Schweiger, and D. Dietrich, editors, *Electronic Commerce– Die Technik*. Huethig Verlag, 2001.

[McM94] J. McMillan. Selling spectrum rights. *Journal of Economic Perspectives*, 8:145–162, 1994.

[MCWG95] A. Mas-Colell, M.D. Whinston, and J.R. Green. *Microeconomic Theory*. Oxford University Press, New York, 1995.

[Mel01] D. Mell. Demand forecasting. Report, Cap Gemini Ernst & Young. Available http://www.cgey.com/energy, 2001.

[Met02] R. Mettu. *Approximation Algorithms for NP-Hard Clustering Problems*. PhD thesis, Department of Computer Science, University of Texas at Austin, 2002.

[Mil89] P. Milgrom. Auctions and bidding: A primer. *Journal of Economic Perspectives*, pages 3–22, 1989.

[MJ01] R. Madlener and E. Jochem. Impacts of market liberaliza-tion on the electricity supply sector: A comparison of the experience in austria and germany. In *Proceedings of the 3rd International Energy Symposium "Dichotomies and Challenges"*, volume 74, pages 131–140. Forschung im Verbund, Oesterreichische Elektrizitaetswirtschafts-AG, 2001.

[MJPL92] S. Minton, M.D. Johnston, A.B. Philips, and P. Laird. Minimizing conflicts: a heuristic repair method for constraint satisfaction and scheduling problems. *Artificial Intelligence*, 58(1-3):161–205, 1992.

[ML04] R. Mailler and V.R. Lesser. Solving distributed constraint optimization problems using cooperative mediation. In *Proceedings of Third International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2004)*, pages 438–445. IEEE Computer Society, 2004.

[MM87] R.P. McAfee and J. McMillan. Auctions and bidding. *Journal of Economics Literature*, 25:699–738, 1987.

[MM96] R.P. McAfee and J. McMillan. Analyzing the airwaves auction. *Journal of Economic Perspectives*, 1(10):159175, 1996.

[Mod03] P.J. Modi. *Distributed Constraint Optimization for Multiagent Systems*. PhD thesis, University of Southern California, 2003.

[Mor93] P. Morris. The breakout method for escaping from local minima. In *Proceedings of the 11th National Conference on Artificial Intelligence*, pages 40–45, 1993.

[Mos91] E. Mosonyi. *Water power development 2A - High-head power plants*. Akadmiai Kiad, 1991.

[MSTM05] P.J. Modi, W.M. Shen, M. Tambe, and M.Yokoo. Adopt: Asynchronous distributed constraint optimization with quality guarantees. *Artificial Intelligence*, 16(1-2):149–180, 2005.

[MSTY03] P. Modi, W.M. Shen, M. Tambe, and M. Yokoo. An asynchronous complete method for distributed constraint optimization. In *Proceedings of the 2nd International Joint Conference on Autonomous Agents and Multi-Agent Systems*, pages 161–168, 2003.

[MT90] S. Martello and P. Toth. *Knapsack Problems, Algorithms and Computer Implementations*. John Wiley and Sons Ltd, England, 1990.

[MTB+04] R.T. Maheswaran, M. Tambe, E. Bowring, J.P. Pearce, and P. Varakantham. Taking dcop to realworld: Efficient complete solutions for multi-event scheduling. In *Proceedings of the 3rd International Joint Conference on Multi-Agent Systems*, 2004.

[Mye79] R.B. Myerson. Incentive compatibility and the bargaining problem. *Econometrica*, 47:61–73, 1979.

[Mye81] R.B. Myerson. Optimal auction design. *Mathematics of Operation Research*, 6:58–73, 1981.

[Nas50] J.F. Nash. Equilibrium points in n-person games. *Proceedings of the National Academy of the USA*, 36(1):48–49, 1950.

[Nis00] N. Nisan. Bidding and allocation in combinatorial auctions. In *ACM Conference on Electronic Commerce*, pages 1–12, 2000.

[Nis06] N. Nisan. Bidding languages. In P. Cramton, Y. Shoham, and R. Steinberg, editors, *Combinatorial Auctions*. MIT Press, 2006.

[NWF78] G. Nemhauser, L. Wolsey, and M. Fisher. An analysis of the approximations for maximizing submodular set functions. *Mathematical Programming*, 14:265–294, 1978.

[OJ96] G.M.P. O'Hare and N.R. Jennings, editors. *Foundations of distributed artificial intelligence*. John Wiley & Sons, Inc., New York, NY, USA, 1996.

[OR94] M.J. Osborne and A. Rubinstein. *A Course in Game Theory*. MIT Press, 1994.

[OS00] S. Oren and P.T. Spiller. High electricity prices in the West: What can be done about it? *Public Utilities Forthnighly*, 138(20), 2000.

[ove91] Regulation (EEC) No 295/91 of 4 February. Official Journal of the European Communities No L 36, 8.2.1991, p. 5, 1991.

[Pal01] P. Palensky. *Distributed Reactive Energy Management*. PhD thesis, Institute of Computer Technology, Vienna University of Technology, 2001.

[Par99] D.C. Parkes. iBundle: An efficient ascending price bundle auction. In *Proceedings of the 1st ACM Conf. on Electronic Commerce*, pages 148–157, 1999.

[Par01] D.C. Parkes. *Iterative Combinatorial Auctions: Achieving Economic and Computational Efficiency*. PhD thesis, Department of Computer and Information Science, University of Pennsylvania, 2001.

[PBB01] C. Preist, A. Byde, and C. Bartolini. Economic dynamics of agents in multiple auctions. In *Proceedings of the 5th International Conference on Autonomous Agents*, pages 545–551, 2001.

[PBS03] Y.K. Penya, A. Bratoukhine, and T. Sauter. Agent-driven distributed manufacturing system for mass customisation. *Integrated Computer-Aided Engineering, IOS Press*, 10(2), 2003.

[PDPR97] P. Palensky, D. Dietrich, R. Posta, and H. Reiter. Demand side management by using lonworks. In *Proceedings of the 2nd IEEE international Workshop on Factory Communication System*, pages 341 – 347, 1997.

[Pea84] J. Pearl. *Heuristics: Intelligent Search Strategies for Computer Problem Solving*. Addison-Wesley, 1984.

[Pen03] Y.K. Penya. Last-generation applied artificial intelligence for energy management in building automation. In *Proceedings of the 5th IFAC International Conference on Fieldbus Systems and their Applications*, pages 79–83, 2003.

[PF05a] A. Petcu and B. Faltings. A scalable method for multiagent constraint optimization. In *Proceedings of the 19th International Joint Conference on Artifcial Intelligence*, Edinburgh, Scotland, Aug 2005.

[PF05b] A. Petcu and B. Faltings. Superstabilizing, fault-containing multiagent combinatorial optimization. In *Proceedings of the 20th National Conference on Artificial Intelligence*, Pittsburgh, Pennsylvania, USA, July 2005.

[PJ05] Y.K. Penya and N.R. Jennings. Combinatorial markets for efficient electricity management. In *Proceedings of the IEEE/WIC/ACM International Conference on Intelligent Agent Technology*, pages 626–632, 2005.

[PJN05] Y.K. Penya, N.R. Jennings, and G. Neumann. An optimal distributed constraint optimisation algorithm for efficient energy management. In *IEEE Proceedings of the 2005 International Conference on Intelligent Agents, Web Technology and Internet Commerce*, 2005.

[PMR02] Y.K. Penya, S. Mahlknecht, and P. Roessler. Lightweight agent platform for high-performance fieldbus nodes. In *Work-in-Progress Proceedings of the 4th IEEE International Workshop on Factory Communication Systems*, pages 9–13, 2002.

[Por99] D. Porter. The effect of bid withdrawal in a multiobject auction. *The Review of Economic Design*, 4:73–97, 1999.

[PPL03] Y.K. Penya, P. Palensky, and M. Lobashov. Requirements and prospects for consumers of electrical energy regarding demand-side management. In *Proceedings of the 2003 Internationale Energiewirtschaftstagung der TU Wien*, pages 101–102, 2003.

[PRD03] P. Palensky, P. Roessler, and D. Dietrich. Heim- und gebaeudeautomatisierung zur effizienzsteigerung in gebaeuden. *Elektrotechnik und Informationstechnik*, 4:104–110, 2003.

[PS02] Y.K. Penya and T. Sauter. Network load imposed by software agents in distributed plant automation. In *Proceedings of the 6th IEEE International Conference on Intelligent Engineering Systems*, pages 339–343, 2002.

[PS04] Y.K. Penya and T. Sauter. Communication issues in multi-agent-based plant automation scheduling. In W. Elmenreich, J.A.T. Machado, and I.K. Rudas, editors, *Intelligent Systems at the Service of the Mankind*, volume 1, pages 135–144. Ubooks, Regensburg, Germany, 2004.

[PU00] D.C. Parkes and L.H. Ungar. Iterative combinatorial auctions: Theory and practice. In *Proceedings of the 17th National Conference on Artificial Intelligence*, pages 74–81, 2000.

[PUF99] D.C. Parkes, L.H Ungar, and D.P Foster. Accounting for cognitive costs in on-line auction design. In P. Noriega and C. Sierra, editors, *Agent Mediated Electronic Commerce (LNAI 1571)*, pages 25–40. Springer-Verlag, 1999.

[PWF$^+$97] V. Parunak, A. Ward, M. Fleischer, J. Sauter, and T. Chang. Distributed component-centered design as agent-based distributed constraint optimization. In *Proceedings of the AAAI Workshop on Constraints and Agents*, pages 93–99, 1997.

[Ram04] S.D. Ramchurn. *Multi-Agent Negotiation using Trust and Persuasion*. PhD thesis, Electronics and Computer Science, University of Southampton, 2004.

[RN02] S.J. Russell and P. Norvig. *Artificial Intelligence: A Modern approach (2^{nd} Edition)*. Prentice Hall, 2002.

[RPH95] M.H. Rothkopf, A. Pekec, and R.M. Harstad. Computationally manageable combinatorial auctions. Technical Report 95-09, 19, 1995.

[RPH98] M.H. Rothkopf, A. Pekec, and R.M. Harstad. Computationally manageable combinatorial auctions. *Management Science*, 44:1131–1147, 1998.

[RSB82] J. Rassenti, V.L. Smith, and R.L. Bulfin. A combinatorial auction mechanism for airport time slot allocation. *Bell Journal of Economics*, (13):402417, 1982.

[RTK90] M. Rothkopf, T. Teisberg, and E. Kahn. Why are vickrey auctions rare? *Journal of Political Economy*, 98:94–109, 1990.

[RZ94] J.S. Rosenschein and G. Zlotkin. *Rules of Encounter*. MIT Press, 1994.

[San91] T.W. Sandholm. A strategy for decreasing the total transportation costs among area-distributed transportation centers. Nordic operations analysis in cooperation (noas): Or in business, turku school of economics, finland, 1991.

[San93a] T.W. Sandholm. An implementation of the contract net protocol based on marginal-cost calculation. In *Proceedings of the 11th National Conference on Artificial Intelligence*, pages 254–262, 1993.

[San93b] T.W. Sandholm. An implementation of the contract net protocol based on marginal cost calculations. In *Proceedings of the 11th National Conference on Artificial Intelligence*, page 256262, 1993.

[San99] T.W. Sandholm. Distributed rational decision. In G. Weiss, editor, *Making Multiagent Systems*, pages 201–258. MIT Press, 1999.

[San00a] T.W. Sandholm. eMediator: a next generation electronic commerce server. In *International Conference on Autonomous Agents*, pages 73–348, 2000.

[San00b] T.W. Sandholm. Issues in computational vickrey auctions. *Internatational Journal of Electronic Commerce*, 3(4):107129, 2000.

[San02] T.W. Sandholm. Algorithm for optimal winner determination in combinatorial auctions. *Artificial Intelligence*, 135(1-2):1–54, 2002.

[San03] T.W. Sandholm. Making markets and democracy work: A story of incentives and computing. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence*, pages 1649–1671, Acapulco Mexico, 2003.

[San06a] T.W. Sandholm. Optimal winner determination algorithms. In P. Cramton, Y. Shoham, and R. Steinberg, editors, *Combinatorial Auctions*. MIT Press, 2006.

[San06b] T.W. Sandholm. The winner determination problem. In P. Cramton, Y. Shoham, and R. Steinberg, editors, *Combinatorial Auctions*. MIT Press, 2006.

[SF02] M.C. Silaghi and B.V. Faltings. Self reordering for security in generalized english auctions. In *Proceedings of the 1st International Joint Conference on Autonomous agents and multiagent systems*, pages 459–460, 2002.

[Sil02] M.C. Silaghi. An algorithm applicable to clearing combinatorial exchanges. Technical Report CS-2002., 2002.

[SN01] P.A. Samuelson and W.D. Nordhaus. *Economics*. McGraw-Hill, 2001.

[SPL+05] M. Stadler, P. Palensky, B. Lorenz, M. Weihs, and C. Roesener. Integral resource optimization networks and their techno-economic constraints. *International Journal on Distributed Energy Systems*, 11(4):199–319, 2005.

[SRSKF91] K. Sycara, S.F. Roth, N. Sadeh-Koniecpol, and M.S. Fox. Distributed constrained heuristic search. *IEEE Transactions on Systems, Man, and Cybernetics*, 21(6):1446–1461, December 1991.

[SS00a] T.W. Sandholm and S. Suri. Improved algorithms for optimal winner determination in combinatorial auctions and generalizations. In *Proceedings of the 17th National Conference on Artificial Intelligence*, pages 90–97, 2000.

[SS00b] T.W. Sandholm and S. Suri. Improved algorithms for optimal winner determination in combinatorial auctions and generalizations. In *Proceedings of the National Conference on Artificial Intelligence*, pages 90–97, 2000.

[SS01] T.W. Sandholm and S. Suri. Market clearability. In *Proceedings of the IJCAI'01*, pages 1145–1151, Seattle WA, 2001.

[SS03] T.W. Sandholm and S. Suri. BOB: Improved Winner Determination in Combinatorial Auctions and Generalizations. *Artificial Intelligence*, 145:33–58, 2003.

[SSGL01a] T.W. Sandholm, S. Suri, A. Gilpin, and D. Levine. CABOB: A Fast Optimal Algorithm for Combinatorial Auctions. In *Proceedings of the*

17th International Joint Conference on Artificial Intelligence, pages 1102–1108, 2001.

[SSGL01b] T.W. Sandholm, S. Suri, A. Gilpin, and D. Levine. Winner determination in combinatorial auction generalizations. In *Proceedings of the AGENTS-2001 Workshop on Agent-Based Approaches to B2B*, pages 35–41, 2001.

[SSGL02] T.W. Sandholm, S. Suri, A. Gilpin, and D. Levine. Winner determination in combinatorial auction generalizations. In *Proceedings of the 1st International Conference on Autonomous Agents and Multiagent Systems*, pages 69–76, 2002.

[SY02] W.M. Shen and M. Yim. Self configurable robots. *IEEE Transactions on Mechatronics*, 7(4), 2002.

[Sys01] BAE Systems. ECM challenge problem.
Available at http://www.sanders.com/ants/ecm.htm, 2001.

[SYS02] SYSTINT. European Interconnection: State of the art 2002. Report of the joint EURELECTRIC-UCTE Working Group “SYSTINT”, Union for the Coordination of Transmission of Energy (UCTE) and Union of the Electricity Industry (Eurelectric), Oct. 2002. Available at http://www.ucte.org, 2002.

[Tam97] M. Tambe. Towards flexibe teamwork. *Journal of Artificial Intelligence Research*, 7:83–124, 1997.

[Ten00] M. Tennenholtz. Some tractable combinatorial auctions. In *Proceedings of the 17th National Conference on Artificial Intelligence*, pages 98–103, 2000.

[UCT02a] UCTE. Union for the Coordination of Transmission of Energy (UCTE) Power Balance Forecasts 2002-2004. Available at http://www.ucte.org, 2002.

[UCT02b] UCTE. Union for the Coordination of Transmission of Energy (UCTE) System Adequacy Forecast 2003-2005 Report. Available at http://www.ucte.org, 2002.

[Uni81] United Nations Energy Resources Development Series 23. Optimization of the utilization of electricity generating plants, United Nations, New York, 1981.

[Vas03] J. Vasconcelos. World Forum on Energy Regulation, Opening Address. Rome, 06.10.2003. Available at www.autorita.energia.it/wfer/vasconcelos-eng.pdf, 2003.

[Vic61] W. Vickrey. Counterspeculation, auctions and competitive sealed tenders. *Journal of Finance*, 16:8–37, 1961.

[Wel93] M.P. Wellman. A market-oriented programming environment and its application to distributed multicommodity flow problems. *Journal of Artificial Intelligence Research*, 1(3):123, 1993.

[Wir94] F. Wirl. *Sind Energiesparmanahmen von Versorgungsunternehmen wirtschaftlich?* Springer Verlag, 1994.

[WJ95] M. Wooldridge and N. R. Jennings. Intelligent agents: Theory and practice. *Knowledge Engineering Review*, 10(2):115–152, 1995.

[Woo02] M. Wooldridge. *An Introduction to MultiAgent Systems*. John Wiley & Sons, Inc., 2002.

[Wur01] P.R. Wurman. Dynamic pricing in the virtual marketplace. *IEEE Internet Computing*, 5(2):36–42, 2001.

[WWW98] P.R. Wurman, M. Wellman, and W. Walsh. The michigan internet auctionbot: A configurable auction server for human and software agents. In *Proceedings of the 2nd International Conference on Autonomous Agents*, pages 301–308, 1998.

[YA96] F. Ygge and H. Akkermans. Power load management as a computational market. In *Proceedings of the 2nd International Conference on Multi-Agent Systems*, pages 393–400, 1996.

[YA97] F. Ygge and H. Akkermans. Duality in multi-commodity market computations. In C. Zhang and D. Lukose, editors, *Proceedings of the 3rd Australian Workshop on Distributed Artificial Intelligence*, pages 65–78, 1997.

[YA98] F. Ygge and H. Akkermans. On resource-oriented multi-commodity market computations. In *Proceedings of the Third International Conference on Multi-Agent Systems*, pages 365–371, 1998.

[YA00] F. Ygge and H. Akkermans. Resource-oriented multi-commodity market algorithms. *Autonomous Agents and Multi-Agent Systems Journal*, 3(1):53–72, 2000.

[YAA+99] F. Ygge, H. Akkermans, A. Andersson, M. Krejic, and E. Bortjes. The homebots system and field tests: A multi- commodity market for predictive load management. In *Proceedings of the 4th International Conference on Practical Application of Intelligent Agents and Multi-Agent Technology*, pages 363–382, London UK, 1999.

[YD91] M. Yokoo and E.H. Durfee. Distributed constraint optimization as a formal model of partially adversarial cooperation. Technical Report CSE-TR-101-91, University of Michigan, 1991.

[YDIK98] M. Yokoo, E.H. Durfee, T. Ishida, and K. Kuwabara. The distributed constraint satisfaction problem: Formalization and algorithms. *IEEE Transactions on Knowledge and Data Engineering*, 10(5):673–685, September 1998.

[YGA96] F. Ygge, R. Gustavsson, and H. Akkermans. Homebots: Intelligent agents for decentralized load management. In *Proceedings of the International Conference and Exhibition on Distribution Automation and Demand Side Management*, pages 597–610, 1996.

[Ygg97] F. Ygge. Power load management as a multi-commodity market. Blekinge university of technology research report 1997:15, 1997.

[Ygg98a] F. Ygge. *Market-oriented programming and its application to power load management*. PhD thesis, Department of Computer Science, Lund University, 1998.

[Ygg98b] F. Ygge. A multi-commodity market approach to power load managemen. Blekinge university of technology research report 1997:15, 1998.

[YH00] M. Yokoo and K. Hirayama. Algorithms for distributed constraint satisfaction: A review. *Autonomous Agents and Multi-Agents Systems*, 3(2):198–212, October 2000.

[YIDK92] M. Yokoo, T. Ishida, E.H. Durfee, and K. Kuwabara. Distributed constraint satisfaction for formalizing distributed problem solving. In *Proceedings of the 12th IEEE International Conference on Distributed Computing Systems*, pages 614–621, 1992.

[Yok93] M. Yokoo. Constraint relaxation in distributed constraint satisfaction problems. In *Proceedings of the 5th International Conference on Tools with Artificial Intelligence*, pages 56–63, 1993.

[Yok95] M. Yokoo. Asynchronous weak-commitment search for solving distributed constraint satisfaction problems. In *Proceedings of the 1st International Conference on Principles and Practice of Constraint Programming*, pages 88–102. Lecture Notes in Computer Science 976, 1995.

[Yok01] M. Yokoo. *Distributed Constraint Satisfaction: Foundations of Cooperation in Multi-agent Systems*. Series on Agent Technology, Springer, Berlin, 2001.

[ZM91] Y. Zhang and A. Mackworth. Parallel and distributed algorithms for finite constraint satisfaction problems. In *Proceedings of the 3rd IEEE Symposium on Parallel and Distributed Processing*, pages 394–397, 1991.

Credits

The Queen pictures in every draughboard are reproductions of the Queen Nefertiti bust at the Egipcian Musuem, Berlin, © Staatliche Museen zu Berlin - Preußischer Kulturbesitz.

Light bulb in Figure 3.1 © S.J. Golding, http://susanthelibrarian.tripod.com

Office building in Figure 3.1 © http://www.zurqui.co.cr

The power station in Figure 3.1 is a gray-scaled reproduction of Pink Floyd's 1977 Album, ”Animals”, depicting the Battersea Power Station (England, UK) taken from the Wikipedia http://en.wikipedia.org/wiki/Animals_(album)

Translations used in the citation:
”Tractatus Logico-Philosophicus” by C.K. Ogden.
”The Aleph'”by N.T. Di Giovanni.
”Faust” by A. S. Kline.
“History” by Aubrey de Selincourt
“Scientific as Vocation” by H.H. Gerth and C. Wright Mills

The poem cited in the Epilogue is “Ithaka” by Konstantinos Kavafis, in the translation by E. Keeley and P. Sherrard.

The rest used material has been produced by the Author or is acknowledged where corresponds.

This work has been written using LaTeX.

List of Tables

List of Figures

List of Acronyms

AI	Artificial Intelligence
ADOPT	Asynchronous Distributed OPTimisation
COBB	Constraint Optimisation by Broadcasting
COB	Constraint Optimisation Problem
CI	Computational Intelligence
CP	Constraint Problem
CSP	Constraint Satisfaction Problem
CDA	Continuous Double Auction
DSM	Demand Side Management
DCOP	Distributed Constraint Optimisation Problem
DCSP	Distributed Constraint Satisfaction Problem
DSO	Distribution System Operator
DRM	Direct Revelation Mechanism
ETSO	European Transmission System Operators
FPSB	First-price sealed-bid Auction
ICM	Incentive Compatible Mechanism
MAS	Multi-Agent System
SCF	Social-Choice Function
SYNCHBB	Synchronous Branch and Bound
TSO	Transmission System Operator
UCTE	Union for the Co-ordination of Transmission of Electricity
UC	Utility Company
VCG	Vickrey-Clarke-Groves Mechanism
VD	Virtual Device
WDP	Winner Determination Problem